

日 本 国 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

1999年 7月23日

出 願 番 号

Application Number:

平成11年特許願第209831号

出 願 人

Applicant(s):

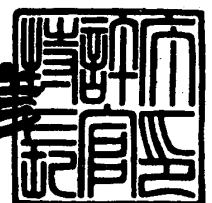
株式会社東芝

CERTIFIED COPY OF
PRIORITY DOCUMENT

2000年 1月28日

特許庁長官
Commissioner,
Patent Office

近 藤 隆 彦



出証番号 出証特2000-3001190

【書類名】 特許願

【整理番号】 A009901940

【特記事項】 特許法第30条第1項の規定の適用を受けようとする特
許出願

【提出日】 平成11年 7月23日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 7/00

【発明の名称】 演算装置及び暗号処理装置

【請求項の数】 14

【発明者】

 【住所又は居所】 東京都府中市東芝町1番地 株式会社東芝府中工場内

 【氏名】 斯波 万恵

【発明者】

 【住所又は居所】 東京都府中市東芝町1番地 株式会社東芝府中工場内

 【氏名】 川村 信一

【特許出願人】

 【識別番号】 000003078

 【氏名又は名称】 株式会社 東芝

【代理人】

 【識別番号】 100058479

 【弁理士】

 【氏名又は名称】 鈴江 武彦

 【電話番号】 03-3502-3181

【選任した代理人】

 【識別番号】 100084618

 【弁理士】

 【氏名又は名称】 村松 貞男

【選任した代理人】

 【識別番号】 100068814

【弁理士】

【氏名又は名称】 坪井 淳

【選任した代理人】

【識別番号】 100092196

【弁理士】

【氏名又は名称】 橋本 良郎

【選任した代理人】

【識別番号】 100091351

【弁理士】

【氏名又は名称】 河野 哲

【選任した代理人】

【識別番号】 100088683

【弁理士】

【氏名又は名称】 中村 誠

【選任した代理人】

【識別番号】 100070437

【弁理士】

【氏名又は名称】 河井 将次

【手数料の表示】

【予納台帳番号】 011567

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 演算装置及び暗号処理装置

【特許請求の範囲】

【請求項 1】 2 の拡大体の多項式基底表現での剰余乗算を実行可能な多倍長の積和演算回路を有する演算装置であって、

前記剰余乗算を乗算処理と剰余算処理とに分割して前記積和演算回路の制御により実行するための制御手段を備えたことを特徴とする演算装置。

【請求項 2】 請求項 1 に記載の演算装置において、

前記積和演算回路は、

前記 2 の拡大体多項式基底における多項式データを乗算するとき、キャリー伝播をしない単精度の乗算回路と、

前記乗算回路による乗算結果を用いて加算を行う倍精度の加算回路とを有し、

前記制御手段は、前記乗算処理のとき、前記乗算回路と前記加算回路を制御することを特徴とする演算装置。

【請求項 3】 請求項 2 に記載の演算装置において、

前記制御手段に制御され、前記剰余算処理のとき、2 つの多項式データの乗算結果を初回の被除多項式データとし、所定の法多項式データを除多項式データとし、前記初回又は 2 回目以降の被除多項式データと前記除多項式データとに基づいて商計算を行い、上位からバス幅と同じビット数の 1 ブロックの商多項式データを立てる商立て回路を有し、

前記制御手段は、前記剰余算処理のとき、前記商立て回路の制御により 1 ブロックの商多項式データが立ったとき、前記乗算回路及び前記加算回路の制御により、前記 1 ブロックの商多項式データと前記除多項式データとの乗算結果を今回の被除多項式データから減らして次の被除多項式データを算出し、前記商立て回路の制御から前記被除多項式データの算出までの処理を繰り返して剰余データを得ることを特徴とする演算装置。

【請求項 4】 請求項 3 に記載の演算装置において、

前記商立て回路は、前記商計算のとき、前記除多項式データの上位 2 ブロックの逆数データと今回の被除多項式データの上位 2 ブロックとを乗算し、この乗算

結果の上位 2 ブロック目を前記 1 ブロックの商多項式データとすることを特徴とする演算装置。

【請求項 5】 請求項 4 に記載の演算装置において、

前記商立て回路は、初回に商多項式データを立てるとき、前記除多項式データの上位 2 ブロックから逆数データを算出してメモリに記憶させ、2 回目以降に商多項式データを立てるとき、前記メモリ内の逆数データを読出して用いることを特徴とする演算装置。

【請求項 6】 請求項 4 又は請求項 5 に記載の演算装置において、

前記商立て回路は、前記逆数データを算出するとき、前記除多項式データの上位 2 ブロックのうち、上位から連続した 0 の数を計数すると、上位から 1 ブロック + 1 ビットの多項式データを最上位ビットを 1 とするように抽出し、この抽出した多項式データの逆数を求め、得られた逆数の最上位ビット側に、最下位ビットが 1 で他のビットが 0 の 1 ブロックの補正データを連結して全体で 2 ブロックのデータを求め、このデータを前記計数した 0 の数だけ上位側にビットシフトさせた結果を前記逆数データとすることを特徴とする演算装置。

【請求項 7】 請求項 1 乃至請求項 6 のいずれか 1 項に記載の演算装置を備え、

前記演算装置による 2 の拡大体の剰余乗算に基づく暗号化又は復号処理を実行することを特徴とする暗号処理装置。

【請求項 8】 請求項 1 乃至請求項 6 のいずれか 1 項に記載の演算装置において、

整数型の単位乗算を実行する場合にはキャリーを伝搬させて単位乗算回路を動作させ、2 の拡大体の単位乗算を実行する場合にはキャリーを伝搬させずに単位乗算回路を動作させるようにしたことを特徴とする演算装置。

【請求項 9】 請求項 1 乃至請求項 6 のいずれか 1 項に記載の演算装置において、

整数型の単位乗算回路と、

前記整数型の単位乗算回路と論理的に隣接して配置された 2 の拡大体の単位乗算回路と、

前記整数型の単位乗算回路を使用するか、前記 2 の拡大体の単位乗算回路を使用するかを選択する選択手段と

を備えたことを特徴とする演算装置。

【請求項 10】 請求項 1 乃至請求項 6 のいずれか 1 項に記載の演算装置において、

整数型の単位乗算回路と、

整数型の単位乗算を実行するか、2 の拡大体の単位乗算を実行するかの選択信号を前記整数型の単位乗算回路に出力する選択制御手段とを備えると共に、

前記整数型の単位乗算回路は、多倍長の積和演算を実行する際に、整数型の単位乗算を実行すべき旨の選択信号を受けたときにはキャリーを伝播し、2 の拡大体の単位乗算を実行すべき旨の選択信号を受けたときにはキャリー伝搬をしないキャリー伝搬制御手段を備え、

前記単位乗算回路におけるキャリー伝搬を制御することにより、整数型乗算と 2 の拡大体の乗算を切替可能に構成されたことを特徴とする演算装置。

【請求項 11】 請求項 10 に記載の演算装置において、

前記キャリー伝搬制御手段は、前記選択信号とキャリーアウト信号を入力とするスイッチによって、1 ビット毎の全加算器におけるキャリーの伝搬制御を行うことを特徴とする演算装置。

【請求項 12】 請求項 10 に記載の演算装置において、

前記キャリー伝搬制御手段は、1 ビット毎の全加算器における 2 入力 a、b の排他的論理和の結果 c を加算結果として出力するか、前記結果 c と入力キャリーとの排他的論理和の結果 d を加算結果として出力するかを切り替える選択手段からなることを特徴とする演算装置。

【請求項 13】 請求項 8 乃至請求項 12 のいずれか 1 項に記載の演算装置において、

前記整数型の乗算を実行する場合にはキャリーを伝搬させて加算を実行し、2 の拡大体の乗算を実行する場合にはキャリーを伝搬させずに加算を実行する加算回路を備えたことを特徴とする演算装置。

【請求項 14】 請求項 8 乃至請求項 13 のいずれか 1 項に記載の演算装置

を備え、

前記演算装置による整数型の演算に基づく暗号化又は復号処理と、前記演算装置による 2 の拡大体の演算に基づく暗号化又は復号処理との双方を切替可能に構成されたことを特徴とする暗号処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

この発明は演算装置及び暗号処理装置、特に例えば IC カードや情報家電製品に実装される暗号処理用コプロセッサ等に用いるのに適した演算装置及び暗号処理装置に関するものである。

【0002】

【従来の技術】

公開鍵暗号の L S I 実装においては、従来から R S A 方式等の整数型の演算を行う暗号方式が主に採用されている。この方式では、桁数の大きい整数についての演算を行う必要があるため、IC カード等に応用する場合には専用コプロセッサが必要とされる。このような専用コプロセッサを実装して暗号処理の多倍長整数演算を実現させる方式については既に多数の実績がある。

【0003】

一方、近年整数型ではなく、2 の拡大体 ($GF(2^m)$: Galois Field (ガロアフィールド)) といわれる代数系の上で構成される暗号系、特に 2 の拡大体の楕円暗号方式が注目されている。

【0004】

この 2 の拡大体演算を用いる暗号方式においても、R S A 方式等の整数型の演算の方式の同様に、扱うビット数を 160 ビット以上と大きくとる必要がある。このため、IC カードのような CPU の演算力が低い装置上でこれを実現しようとすると比較的処理時間がかかる。したがって、専用のハードウェア (コプロセッサ) を用いて高速化したいという要求がある。

【0005】

このように R S A 方式であっても、楕円暗号方式であっても、IC カード等に

において高速な暗号処理を実現させるためには専用の演算用コプロセッサを用意する必要がある。

【0006】

図23は暗号処理用のコプロセッサを含むICカード用LSIのブロック図である。

【0007】

同図に示す構成要素を含む専用LSIが例えばICカードに組み込まれる。このLSIにおいては、CPU、RAM、ROM、EEPROMが1チップに収容され、コプロセッサはRAM、演算部、制御部から構成される。コプロセッサはCPUの制御下にあつて多倍長のべき乗剰余演算や四則演算といった公開鍵暗号の基本演算に関しCPUを補助する。つまり、暗号の基本演算を行う部分は図10におけるコプロセッサであり、この部分がどのように構成されているかが問題となる。

【0008】

図24は図23に示すLSIにおけるコプロセッサ部分の構成例を示すブロック図である。

【0009】

RSA方式では整数型の演算が行われるため、このコプロセッサは図24に示すように整数型の乗算器として構成されている。

【0010】

一方、楕円暗号方式のLSIを作成する場合、その全体的な構成は図23に示すLSIと同様あるいはこれと類似する構成のものとなるが、コプロセッサ部分における演算は整数型の演算でなく、2の拡大体演算を行うコプロセッサを用意する必要がある。

【0011】

図25は多項式基底の2の拡大体演算を行うコプロセッサのハードウェア構成例を示すブロック図である。

【0012】

同図は、「楕円暗号のハードウェア実装，SCIS'98-10.1.C」に

て発表された特殊な既約多項式を用いる円分体といわれる 2 の拡大体の一種の演算装置を示している。この演算装置は、2 の拡大体上の加算、二乗、乗算及び逆元演算を実行できる構成を備えており、これによって楕円曲線上の点の演算に必要な 2 の拡大体の演算を実行する。このような演算装置を IC 化すれば、図 23 の LSI に適用し得る 2 の拡大体演算用のコプロセッサになる。

【0013】

ここで、加算回路および二乗算回路は m 個の EX-OR で構成され、乗算回路 81 は図 26 に示す回路構成で実現する。

【0014】

図 26 は円分体といわれる 2 の拡大体の乗算回路を示す図である。

【0015】

この乗算回路 81 は、 m ビットの入力レジスタ A、B を持ち多項式 $a(x)$ の係数を入力レジスタ A に固定値として入力し、入力レジスタ B からは多項式 $b(x)$ の係数を最上位ビットから 1 クロック毎にシフトしながら演算していく。同図におけるブロック D はフィードバックレジスタを構成するフリップフロップである。 m 回シフトした時点での各ブロック D の値が出力レジスタ C に読み出され、 $a(x) * b(x)$ の演算結果となる。

【0016】

図 24 と図 26 夫々の回路を見比べてもわかるように、乗算と一口に言っても整数型乗算と多項式基底の 2 の拡大体乗算とでは、その乗算を実行するためのアーキテクチャーが全く異なる。したがって、従来は暗号方式毎にハードウェアを構成しようとする試みがなされてきた。

【0017】

一方、楕円曲線暗号の基本演算における 2 の拡大体の剰余乗算には、拡大体 $G_F(q^m)$ 上の多項式 $f(x)$ による割り算回路としての線形フィードバックシフトレジスタ (LFSR) を用いた演算装置が広く用いられている。なお、法多項式 $f(x)$ は、次式の通りである。

【0018】

$$f(x) = f_m x^m + f_{m-1} x^{m-1} + \cdots + f_1 x + f_0, \quad f_m = 1$$

図27は係る線形フィードバックシフトレジスタLFSRの構成を示すブロック図である。このLFSR90は、入力側からEX-ORの加算器91₁～91_mと1クロックの遅延素子（以下、レジスタという）92₁～92_mとが交互に縦続接続されており、m個目のレジスタ92_mから得られる出力が抽出されて夫々係数器93₁～93_mを介して個別にm個の加算器91₁～91_mにフィードバックされる構成となっている。

【0019】

このLFSR90は、単位時間（クロック）毎に動作する。なお、シフトレジスタにおいて、動作用のクロックパルスを1クロック進めることをシフトするといひ、シフトレジスタに組込まれるレジスタ92₁～92_mの数mをシフトレジスタの段数という。

【0020】

ここで、 $q=2$ のとき、各レジスタ92₁～92_mは、1ビットのフリップフロップが適用可能である。各係数器93₁～93_mは、“1”又は“0”を乗じるものであり、1を乗じる場合には結線されるが、0を乗じる場合には結線されない。また、各加算器91₁～91_mは、2入力のEX-ORが使用されている。

【0021】

このようなLFSR90では、入力側（左側）から被除多項式の係数が高次の項から順に入力されると、出力側（右側）から商多項式の係数が高次の項から順に出力される。ここで、被除多項式の0次の項を入力し終えたときの各レジスタ（フリップフロップ）92₁～92_mの内容が剰余多項式の係数となる。

【0022】

しかしながら、以上のLFSR90を用いた演算装置では、拡大次数mのビット数と同数のレジスタ92₁～92_mを必要とし、レジスタ92₁～92_mの構成が拡大次数mにより制約を受けている。このため、拡大次数mが増加すると、演算装置ごと作り直す必要が生じてしまう。

【0023】

【発明が解決しようとする課題】

上記したように、楕円暗号方式は現在注目されてはいるものの、現状では RSA 暗号方式が未だ主流であるため、楕円暗号方式を用いる IC カードにおいても RSA 暗号にも対応させたいという要請が強い。

【0024】

ここで、従来の整数型の暗号と 2 の拡大体の暗号を同一 IC カードに実装しようとした場合、上記した従来技術の延長ではそれぞれに対応するコプロセッサを搭載する必要がある。しかしながら、2 つのコプロセッサを搭載したのでは、面積制約の大きい IC カードにおいてそのチップ面積を圧迫するという問題が生じる。

【0025】

一方、2 の拡大体の剰余乗算においては、拡大次数 m の増加により、演算装置ごと作り直す必要があるというハードウェア的な制限がある。

【0026】

本発明はこのような実情を考慮してなされたもので、2 の拡大体の拡大次数 m を増加しても、装置本体を作り直さずに演算を実行し得る演算装置及び暗号処理装置を提供することを目的とする。

【0027】

また、本発明の他の目的は、最小のアーキテクチャを追加するだけで整数型の演算に加えて 2 の拡大体上の演算をも実行できる演算装置及び暗号処理装置を提供することにある。

【0028】

【課題を解決するための手段】

上記課題を解決するために、請求項 1 に対応する発明は、2 の拡大体の多項式基底表現での剰余乗算を実行可能な多倍長の積和演算回路を有する演算装置であって、剰余乗算を乗算処理と剰余算処理とに分割して積和演算回路の制御により実行するための制御手段を備えた演算装置である。

【0029】

本発明はこのような手段を設けたので、剰余算処理の際に線形フィードバックシフトレジスタではなく、多倍長の積和演算回路が演算を行うので、1 以上の任

意の拡大次数を使用でき、2の拡大体の拡大次数を増加しても、装置本体を作り直さずに演算を実行することができる。

【0030】

次に、請求項2に対応する発明は、請求項1に対応する演算装置において、積和演算回路としては、2の拡大体多項式基底における多項式データを乗算するとき、キャリー伝播をしない単精度の乗算回路と、乗算回路による乗算結果を用いて加算を行う倍精度の加算回路とを有し、制御手段としては、乗算処理のとき、乗算回路と加算回路を制御する演算装置である。

【0031】

本発明はこのような手段を設けたので、請求項1と同様の作用を容易且つ確実に奏することができる。

【0032】

次に、請求項3に対応する発明は、請求項2に対応する演算装置において、制御手段に制御され、剰余算処理のとき、2つの多項式データの乗算結果を初回の被除多項式データとし、所定の法多項式データを除多項式データとし、初回又は2回目以降の被除多項式データと除多項式データとに基づいて商計算を行い、上位からバス幅と同じビット数の1ブロックの商多項式データを立てる商立て回路を有し、制御手段としては、剰余算処理のとき、商立て回路の制御により1ブロックの商多項式データが立ったとき、乗算回路及び加算回路の制御により、1ブロックの商多項式データと除多項式データとの乗算結果を今回の被除多項式データから減らして次回の被除多項式データを算出し、商立て回路の制御から被除多項式データの算出までの処理を繰り返して剰余データを得る演算装置である。

【0033】

この演算装置では、1ブロックの商多項式データと除多項式データとの乗算結果が毎回 $(m+1)$ ブロックになる。

また、この乗算結果を今回の被除多項式から減算（＝加算）して、 $(2m-1 * n)$ ブロックの次回の被除多項式データを算出し（ n は乗算回数）、すなわち前回の被除多項式データを1ブロックずつ減らしていく。

【0034】

本発明はこのような手段を設けたので、請求項1又は請求項2と同様の作用に加え、ハードウェアの特性を生かして剰余算と商計算の効率化を図ることができる。

【0035】

次に、請求項4に対応する発明は、請求項3に対応する演算装置において、商立て回路としては、商計算のとき、除多項式データの上位2ブロックの逆数データと今回の被除多項式データの上位2ブロックとを乗算し、この乗算結果の上位2ブロック目を1ブロックの商多項式データとする演算装置である。

【0036】

本発明はこのような手段を設けたので、請求項3と同様の作用に加え、得られた商多項式のうち、有効数字の部分を抽出できるので、演算精度の最適化を図ることができる。

【0037】

次に、請求項5に対応する発明は、請求項4に対応する演算装置において、商立て回路としては、初回に商多項式データを立てるとき、除多項式データの上位2ブロックから逆数データを算出してメモリに記憶させ、2回目以降に商多項式データを立てるとき、メモリ内の逆数データを読み出して用いる演算装置である。

【0038】

本発明はこのような手段を設けたので、同じ法多項式下での重複する剰余算の実行時に、逆数データをメモリから読み出して商を立てるので、2回目の商計算以降は逆数データの算出時間を省略でき、もって、2の拡大体演算の処理時間を短縮することができる。また、逆数データを事前に計算できるので、2の拡大体の剰余乗算を、乗算と加算とを行う積和演算回路のみを用いて実現することができる。

【0039】

次に、請求項6に対応する発明は、請求項4又は請求項5に対応する演算装置において、商立て回路としては、逆数データを算出するとき、除多項式データの上位2ブロックのうち、上位から連続した0の数を計数すると、上位から1ブロック+1ビットの多項式データを最上位ビットを1とするように抽出し、この抽

出した多項式データの逆数を求め、得られた逆数の最上位ビット側に、最下位ビットが1で他のビットが0の1ブロックの補正データを連結して全体で2ブロックのデータを求め、このデータを前記計数した0の数だけ上位側にビットシフトさせた結果を逆数データとする演算装置である。

【0040】

本発明はこのような手段を設けたので、一般的な多倍長の整数型の除算で用いられる単精度除算を用いたKnuthのアルゴリズム（文献：Knuth, D. E. The Art of Computer Programming, Vol.2, Reading, Mass.: Addison Wesley, 2nd edition, (1981)）による除数の正規化や近似商の補正処理、商や剰余といった演算結果の逆正規化の処理を行わない観点から、予め補正した値を逆数データとしたので、ビットシフトの回数を減らすことができ、演算装置を最適化することができる。

【0041】

次に、請求項7に対応する発明は、請求項1乃至請求項6のいずれか1項に対応する演算装置を備え、演算装置による2の拡大体の剰余乗算に基づく暗号化又は復号処理を実行する暗号処理装置である。

【0042】

本発明はこのような手段を設けたので、楕円曲線暗号等の2の拡大体の剰余乗算に基づく暗号化又は復号処理を実行することができる。

【0043】

次に、請求項8に対応する発明は、請求項1乃至請求項6のいずれか1項に対応する演算装置であって、整数型の単位乗算を実行する場合にはキャリーを伝搬させて単位乗算回路を動作させ、2の拡大体の単位乗算を実行する場合にはキャリーを伝搬させずに単位乗算回路を動作させるようにした演算装置である。

【0044】

本発明はこのような手段を設けたので、請求項1乃至請求項6のいずれかの作用に加え、最小のアーキテクチャを追加するだけで整数型の演算に加えて2の拡大体上の演算をも実行することができる。

【0045】

次に、請求項 9 に対応する発明は、請求項 1 乃至請求項 6 のいずれか 1 項に対応する演算装置において、整数型の単位乗算回路と、整数型の単位乗算回路と論理的に隣接して配置された 2 の拡大体の単位乗算回路と、整数型の単位乗算回路を使用するか、2 の拡大体の単位乗算回路を使用するかを選択する選択手段とを備えた演算装置である。

【0046】

本発明はこのような手段を設けたので、請求項 1 乃至請求項 6 のいずれかの作用に加え、2 の拡大体の単位乗算回路を追加するだけで整数型の乗算と 2 の拡大体の乗算の双方を実行することができる。

【0047】

次に、請求項 10 に対応する発明は、請求項 1 乃至請求項 6 のいずれか 1 項に対応する演算装置において、整数型の単位乗算回路と、整数型の単位乗算を実行するか、2 の拡大体の単位乗算を実行するかの選択信号を整数型の単位乗算回路に出力する選択制御手段を備えるとともに、整数型の単位乗算回路は、多倍長の積和演算を実行する際に、整数型の単位乗算を実行すべき旨の選択信号を受けたときにはキャリーを伝播し、2 の拡大体の単位乗算を実行すべき旨の選択信号を受けたときにはキャリー伝搬をしないキャリー伝搬制御手段を備え、単位乗算回路におけるキャリー伝搬を制御することにより、整数型乗算と 2 の拡大体の乗算を切替可能に構成された演算装置である。

【0048】

本発明はこのような手段を設けたので、請求項 1 乃至請求項 6 のいずれかの作用に加え、キャリー伝搬制御手段を追加するだけで整数型の乗算と 2 の拡大体の乗算の双方を実行することができる。

【0049】

次に、請求項 11 に対応する発明は、請求項 10 の演算装置において、キャリー伝搬制御手段は、選択信号とキャリーアウト信号を入力とするスイッチによって、1 ビット毎の全加算器におけるキャリーの伝搬制御を行う演算装置である。

【0050】

本発明はこのような手段を設けたので、選択信号とキャリーアウト信号を入力

とするスイッチにより請求項 10 に係る発明を実現させることができる。

【0051】

次に、請求項 12 に対応する発明は、請求項 10 の演算装置において、キャリー伝搬制御手段は、1 ビット毎の全加算器における 2 入力 a, b の排他的論理和の結果 c を加算結果として出力するか、結果 c と入力キャリーとの排他的論理和の結果 d を加算結果として出力するかを切り替える選択手段からなる演算装置である。

【0052】

本発明はこのような手段を設けたので、選択手段により請求項 10 に係る発明を実現させることができる。

【0053】

次に、請求項 13 に対応する発明は、請求項 8～12 の演算装置において、整数型の乗算を実行する場合にはキャリーを伝搬させて加算を実行し、2 の拡大体の乗算を実行する場合にはキャリーを伝搬させずに加算を実行する加算回路を備えた演算装置である。

【0054】

本発明はこのような手段を設けたので、請求項 8～12 のいずれかの作用に加え、積和演算における加算部分についても整数型の乗算と 2 の拡大体の乗算との双方を確実に実行することができる。

【0055】

次に、請求項 14 に対応する発明は、請求項 8～13 の何れかの演算装置を備え、演算装置による整数型の演算に基づく暗号化又は復号処理と、演算装置による 2 の拡大体の演算に基づく暗号化又は復号処理との双方を切替可能に構成された暗号処理装置である。

【0056】

本発明はこのような手段を設けたので、請求項 8～13 のいずれかの作用に加え、RSA 暗号等の整数型の演算に基づく暗号と、楕円曲線暗号等の 2 の拡大体の演算に基づく暗号の双方の処理を行うことができる。

【0057】

【発明の実施の形態】

以下、本発明の各実施形態について図面を用いて説明する。

(第1の実施形態)

図1は本発明の第1の実施形態に係る演算装置の構成例を示すブロック図である。

コプロセッサ1として構成される本実施形態の演算装置は、整数型乗算及び2の拡大体乗算の双方の演算が可能な多倍長積和乗算装置であり、この乗算処理を仕方を制御することにより、加算、二乗あるいは逆元等の他の演算を実行するものである。また、本演算装置がLSI等に組み込まれることによってRSA暗号及び楕円暗号の双方が実現可能な暗号処理装置が構成される。ここで組込対象となるLSIは例えば図23に示すような装置である。

【0058】

このコプロセッサ1において演算部4は制御部5によってコントロールされ、演算途中のデータを格納するメモリ2から接続される32ビットのデータバス3からデータを入出力するようになっている。

【0059】

データバス3からの入力データはバッファZ、Y、Xに格納され、データバス3への出力データはバッファRに格納されるようになっている。

【0060】

入力データX及びYは乗算対象となるデータであり、このうちデータYは一度に多数桁の乗算となるのを回避するために所定桁毎に分割されたデータとして入力される。一方、データZは乗算を複数回に分けて実行するために生じる途中結果であり、これをXYの乗算結果に足し、さらにその和の結果にキャリーCと言われる桁上がり部分を足して1サイクルの乗算が終了する。その結果からキャリーを除いたデータRがバッファRを介してデータバス3に出力され、次のサイクルの演算にデータZとして使用される。このサイクルを複数回繰り返すことにより多倍長整数乗算あるいは2の拡大体乗算（厳密には後述のc'）の乗算が実現される。

【0061】

また、コプロセッサ 1 は、上記演算を実現するために、バッファ X、Y、Z、R の他、整数型乗算回路 11、2 の拡大体上乘算回路 12、セクタ 13、加算回路 14、加算回路 15、キャリー保持部 16 及び制御部 5 を備えている。

【0062】

整数型乗算回路 11 は、バッファ X 内のデータ X とバッファ Y 内のデータ Y とを整数型乗算し、その結果をセクタ 13 に出力する。

【0063】

2 の拡大体上乘算回路 12 は、バッファ X 内のデータ X とバッファ Y 内のデータ Y とにより 2 の拡大体上乘算の一部 (c') を実行し、その結果をセクタ 13 に出力する。

【0064】

セクタ 13 は、制御部 5 からの信号 S1 に従って、整数型乗算回路 11 又は 2 の拡大体上乘算回路 12 からの出力の何れかを加算回路 14 に出力する。

【0065】

加算回路 14 は全加算器からなり、バッファ Z 内のデータ Z とセクタ出力を加算して加算回路 15 に出力する。この加算回路 14 においては、整数型の加算と 2 の拡大体の加算との切替が制御信号 S1 に従って行われるようになっている。なお、この加算切替については後述する。

【0066】

加算回路 15 は、加算回路 14 の出力にキャリー保持部 16 に保持されたキャリー C を加算し上位 32 ビットを次のキャリー C としてキャリー保持部 16 に出力し、下位 8 ビットをこのサイクルの演算結果であるデータ R としてバッファ R に出力する。なお、加算回路 15 においても、制御信号 S1 により、整数型の加算と 2 の拡大体の加算との切替が行われるようになっている。

【0067】

キャリー保持部 16 は、加算回路 15 から出力されたキャリー C を保持し、次の演算サイクルにおいて保持したキャリー C を加算回路 15 に与える。

【0068】

制御部 5 は、整数演算制御部 21 と 2 の拡大体演算制御部 22 からなり、これ

らの何れかのコマンド群に従って演算部を制御する。このコマンド切り替えは、外部のCPU（例えば図23に示すCPU）からの指示によって行われる。

【0069】

整数演算制御部21は、演算部4を多倍長整数演算型の乗算器として動作するように制御するものである。このために、制御信号S1によりセクタ13が整数型単精度乗算器11の出力を加算回路14に出力するように制御するとともに、加算回路14及び15を整数型加算回路として動作するように制御する。さらに、整数型乗算器として演算部4の動作を制御することで他の四則演算などの演算処理の実行する。

【0070】

また、2の拡大体演算制御部22は、演算部4を2の拡大体乗算器として動作するように制御するものである。このために、制御信号S1によりセクタ13が2の拡大体型単精度乗算器11の出力を加算回路14に出力するように制御するとともに、加算回路14及び15を2の拡大体型加算回路として動作するように制御する。さらに、2の拡大体型乗算器として演算部4の動作を制御することで加算、二乗算を実現する。

【0071】

なお、制御部5からは、上記した各処理を実現するため、制御信号S2を出力して各部を制御する。

【0072】

次に、以上のように構成された本実施形態における演算装置の動作について説明する。

【0073】

この演算装置（コプロセッサ1）は、整数型の乗算装置に乗算回路12、セクタ13等を組み込むことにより、2の拡大体の乗算装置としての処理を実現可能とするものである。ここで、2の拡大体では、以下に示すように $m-1$ 次の多項式を m ビットのベクトル表現で表すことができる。

【0074】

$$a(x) = a_{m-1} x^{m-1} + a_{m-2} x^{m-2} + \dots + a_1 x + a_0 \quad \dots (1)$$

$$\begin{aligned}
 &= [a_{m-1}, \dots, a_1, a_0] \\
 b(x) &= b_{m-1} x^{m-1} + b_{m-2} x^{m-2} + \dots + b_1 x + b_0 \quad \dots (2) \\
 &= [b_{m-1}, \dots, b_1, b_0]
 \end{aligned}$$

ここで、2の拡大体の乗算はGF(2^m)上のm次の規約多項式f(x)をモジュラスとする剰余乗算である。また、2の拡大体の二つの元a(x)とb(x)の積c(x)は、次のように定義されている。

【0075】

$$\begin{aligned}
 c(x) &= a(x) \cdot b(x) \bmod f(x) \quad \dots (3) \\
 &= \sum a_k \cdot x^k \cdot b(x) \bmod f(x) \\
 &= c_{m-1} x^{m-1} + c_{m-2} x^{m-2} + \dots + c_1 x + c_0 \\
 &= [c_{m-1}, \dots, c_1, c_0]
 \end{aligned}$$

また、法多項式f(x)は、次式で表せる。

$$\begin{aligned}
 f(x) &= f_m x^m + f_{m-1} x^{m-1} + \dots + f_1 x + f_0 \quad \dots (4) \\
 &= [f_m, f_{m-1}, \dots, f_1, f_0]
 \end{aligned}$$

2の拡大体の多項式の乗算は、図26に示すように乗数のサイクルシフトによるシフトレジスタを構成し、mサイクルシフト後の剰余多項式を乗算結果とするのが一般的であるが、本実施形態では整数型の暗号処理LSIで広く使われている多倍長の積和演算回路に若干の変更を加えて処理する。

【0076】

なお、制御部5からの制御信号S1により、コプロセッサ1が整数型の演算装置として動作するときには、同演算装置は多倍長積和演算回路として機能している。この多倍長積和演算回路において、制御信号S1による切替により、2の拡大体上乘算回路12において2の拡大体の乗算の一部分である(5)式が計算される。

【0077】

$$c'(x) = a(x) \cdot b(x) \quad \dots (5)$$

なお、2の拡大体上乘算回路12ではc'を計算する段階においては(6)式における「c(x)' mod f(x)」の部分は計算されない。すなわちc'自体は、制御信号S1により乗算回路12及び加算回路14、15を切り替える

のみで、整数型乗算における2つの数の積と全く同様に演算される。

【0078】

なお、 $c'(x) = a(x) \cdot b(x)$ において m ビットの乗数、被乗数は 32 ビットに分割されてメモリから読み出され、演算結果は 32 ビット毎にメモリに書き込まれる。この時、最終的な演算結果は $2m$ ビットとなる。

【0079】

整数型乗算回路 11 による整数演算と 2 の拡大体上乘算回路 12 による 2 の拡大体多項式演算の違いは、桁上がりの有無である。整数演算では足し算の論理式は

$$0 + 0 + \text{Carry} (=0) = 0, \text{Carry} = 0$$

$$1 + 0 + \text{Carry} (=0) = 1, \text{Carry} = 0$$

$$1 + 1 + \text{Carry} (=0) = 0, \text{Carry} = 1$$

という様に下位ビットのキャリーを考慮した演算をしなければならないのに対し、2 の拡大体の代数系においては、各ビットが多項式における次数の係数を示しているため異なる次数への桁上がりを考慮しなくてもよい。

【0080】

このことに着目して本実施形態では整数型演算器（乗算器や加算器）において、キャリー伝播を許す通常モードと、キャリー伝播を実行しないモードを切り替えて使えるようにしているのである。ここでキャリー伝播を許さない（実行しない）モードは 2 の拡大体演算を行うのに用いられる。なお、キャリー伝播のモードを切り替えのために追加すべき回路は全体の回路規模に比べわずかである。

【0081】

図 2 は $c'(x) = a(x) \cdot b(x)$ を実現するための 4×4 ビットの単位乗算の回路構成例を示す図である。

【0082】

同図の単位演算装置を 8×32 ビット構成にしたものが図 1 における 2 の拡大体上乘算回路 12 である。なお、図 2 (b) の回路は同図 (a) の回路の入力部分 29 を示すものである。

【0083】

一方、図 3 は整数型乗算を実現するための 4 * 4 ビットの単位乗算の回路構成例を示す図である。

【0084】

同図の単位演算装置を 8 * 32 ビット構成にしたものが図 1 における整数型乗算回路 11 である。なお、図 3 (a) に用いられる全加算器 FA の構成は図 3 (c) に示され、さらに図 3 (c) に示す全加算器 FA のキャリー 31 の構成が同図 (d) に示されている。また、図 3 (b) の回路は同図 (a) の回路の入力部分 30 を示すものである。

【0085】

本実施形態の演算装置では、2 の拡大体上乘算回路 12 と整数型乗算回路 11 とが論理的に隣接して配置されており、制御部 5 の 2 の拡大体演算コマンドから生成される制御信号 S1 により整数型、2 の拡大体型のいずれかの乗算回路 11, 12 が選択されて処理が行われる。

【0086】

セクタ 13 の出力は次段の加算回路 14 に入力される。ここで $Z + (Y * X)$ 加算回路 14 は 40 ビットのデータ $(Y * X)$ と 8 ビットのデータ Z の全加算器だが、ここでも前述の制御信号により各ビットの加算結果のキャリーを次段へ伝播しないスイッチを付加することにより 2 の拡大体の加算が実現される。

【0087】

図 4 は本実施形態におけるコプロセッサに用いられるキャリー制御機能付きの 4 ビットのリップルキャリー型全加算器の構成例を示すブロック図である。

【0088】

このような構成の全加算器を、40 ビットデータと 8 ビットデータとの加算が可能となるように拡張したものが図 1 の加算回路 14 である。

【0089】

また、図 4 の回路において、各全加算器 32 の間にはスイッチ 33 が設けられ、キャリーの伝搬を制御できるようになっている。

【0090】

図 5 は本実施形態の加算回路に用いられる全加算器及びキャリー制御スイッチ

の構成例を示す図である。

【0091】

この全加算器 32 及びスイッチ 33 は、1 ビット分のキャリー制御機能付き全加算器 42 を構成している。ここで、全加算器 32 は、図 3 (c) に示す全加算器 FA と同様に構成され、全加算器 32 内のキャリー 31 は図 3 (d) に示すキャリーと同様に構成されている。

【0092】

また、全加算器 32 間のキャリー伝搬ラインに設けられたスイッチ 33 は制御部 5 からの制御信号 S1 によって制御され、整数型演算を行うときには接続され、2 の拡大体演算を行うときには遮断される。

【0093】

以上のように構成された加算回路 14 からの出力 ($Z + (Y * X)$) は加算回路 15 に引き渡される。

【0094】

すなわち、演算ブロック最終段の $C + Z + (Y * X)$ 加算回路 15 によって、乗算結果の 40 ビットの下位 8 ビットがデータ R として出力され、上位 32 ビットが次のサイクルの $Z + (Y * X)$ に足し込まれる。

【0095】

ここで、加算回路 15 は加算回路 14 と同様に、前述の制御信号 S1 により制御される図 4 に示すキャリー制御機能付き全加算器であるので、整数型では LSB に桁あわせをした全加算器として整数型加算が実行され、2 の拡大体演算では 2 の拡大体加算が実行される。

【0096】

加算回路 15 の出力データ R はデータバス 3 を介して一旦外部のメモリ 2 に出力され、再びデータ Z となってコプロセッサ 1 内に戻り整数型乗算若しくは 2 の拡大体上の乗算が継続され、必要なサイクル数だけ繰り返されて乗算結果が得られる。

【0097】

ここで 2 の拡大体の乗算コマンドでは、(5) 式の結果が得られるが、2 の拡

大体乗算は(6)式に示す定義通り、既約多項式 $f(x)$ をモジュラスとする剰余演算によって完結する。剰余演算は割り算の筆算同様、被除数の上位桁から商を立て現在の商と除数をかけたものから現在の被除数を引く(2の拡大体では減算は加算と同じ)処理を必要なサイクル数だけ繰り返せばよく、2の拡大体の乗算コマンドと加算コマンドを実行することによって実現できる(詳細は第3の実施形態で述べる)。2の拡大体の二乗算は乗算と同じ処理で実現でき、逆元計算は、乗算と二乗算を相互に繰り返すことにより実現できる。

【0098】

一例として、2の拡大体の加算コマンドに従って演算部4が2の拡大体の加算装置として機能する場合を説明する。

【0099】

2の拡大体上の加算は、通常が多項式の加算と同じで、同じ次数の係数同士の足し算を行う。

【0100】

$$\begin{aligned} c(x) &= a(x) + b(x) \quad \dots (7) \\ &= [a_{m-1} + b_{m-1}, a_{m-2} + b_{m-2}, \dots, a_0 + b_0] \end{aligned}$$

このとき、各次数の係数の和は $0+0=1+1=0$ 、 $0+1=1+0=1$ となり、整数型加算のようにキャリーは発生しない。従って、2の拡大体での加算は、一般には m 個の EX-OR で実装できることになる。

【0101】

整数型の乗算装置において加算は $c = b + a * 1$ として扱えるので、本実施形態における2の拡大体の加算もこのアルゴリズムをそのまま利用し、 $c(x) = b(x) + a(x) * 1$ として実行する。この演算は加算回路14、15に図4の全加算器が用いられていることから、制御信号 $S1$ の切替で実現できる。

【0102】

また、制御信号 $S1$ による切替でコプロセッサ1は図24に示すコプロセッサと同様な機能を持つ回路となり、整数型演算も実現される。

【0103】

上述したように、本発明の実施の形態に係る演算装置は、整数型乗算装置に、

整数型乗算の単位乗算装置と回路構成の似ている2の拡大体乗算の単位演算装置とを設け、整数型の乗算コマンドに2の拡大体演算コマンドとを追加し、2の拡大体演算コマンドから生成される制御信号により制御されるセクタと、全加算器の各ビットのキャリーの伝播を制御するスイッチの追加するようにしたので、従来型のシフトレジスタによるシーケンシャルな2の拡大体の乗算装置を用いることなく整数および2の拡大体演算の両方を実行することができる。

【0104】

したがって、従来からある整数型の演算器への追加拡張機能として、ごく少ない命令と回路の追加することにより、多倍長の積和演算回路で2の拡大体の加算、乗算を実行することが可能な公開鍵暗号処理用アクセラレータを提供することができる。なお、本実施形態を実現するのに、必要な回路追加の量は全体の回路規模に比べてわずかである。

【0105】

本実施形態の暗号処理装置によれば、暗号処理用コプロセッサとして、整数型のRSA方式に加え2の拡大体の楕円暗号方式も処理できる豊富な機能をもつLSIを特に実装面積を増大させることなく提供できる。したがって、ICカードのような実装可能面積の少ない装置において、RSA、楕円暗号の双方を処理できる暗復号装置を実現させることができる。

【0106】

(変形例1)

本変形例では、図4に示す加算回路14、15を構成するキャリー制御機能付き全加算器について説明する。

【0107】

図6はキャリー制御機能付き全加算器の変形例を示す図である。

【0108】

このキャリー制御機能付き全加算器43は、スイッチ33と全加算器32から構成される点で図5の回路と共通する。しかし、図5の回路ではキャリー31の出力側にスイッチ33が設けられているのに対し、図6の回路ではキャリー31の入力側にスイッチ33が設けられている。

【0109】

(変形例2)

本変形例では、更に他のキャリー制御機能付き全加算器について説明する。

【0110】

図7はキャリー制御機能付き全加算器の他の変形例を示す図である。

【0111】

このキャリー制御機能付き全加算器44は、加算結果の出力選択を制御することによりキャリー制御を行う。すなわちスイッチ33'はセクタで構成され、このセクタは制御信号S1に基づき、EXOR35又はEXOR36の出力を選択する。これを複数個連結したリップルキャリー型加算器は、制御信号S1によりキャリー伝播の有無を制御できる。

【0112】

図7の制御信号S1を2の拡大体演算コマンドによる制御信号とするとS1が“1”のときaとbのEXOR35の出力が演算結果となり、2の拡大体の加算装置として機能し、S1が“0”のとき全加算器の出力が演算結果となって整数型の加算装置として機能する。

【0113】

(第2の実施形態)

図8は本発明の第2の実施形態に係る演算装置の構成例を示すブロック図であり、図1と同一部分には同一符号を付して説明を省略し、ここでは異なる部分についてのみ述べる。なお、以下の各実施形態も同様に重複した説明を省略する。

【0114】

この演算装置であるコプロセッサ1'は、図1における整数型乗算回路11、2の拡大体上乘算回路12及びセクタ13に代えて乗算回路41を備える他、第1の実施形態と同様に構成されている。

【0115】

この乗算回路41は、制御部5からの制御信号S1によって整数型乗算と2の拡大体上乘算((6)式のc'のみ)を切り替えるようになっている。

【0116】

図9は本実施形態の乗算回路を実現するための4*4ビットの単位乗算の回路構成例を示す図である。なお、現実の乗算回路41は、同図の単位演算装置を8*32ビット構成にしたものである。また、図9(b)の回路は同図(a)の回路の入力部分29を示すものである。

【0117】

この乗算回路41は、図9(a)に示すように、全加算器として図5に示すキャリー制御機能付き全加算器42を用いているので、制御信号S1に従ってキャリー伝搬の有無を制御できる。したがって、2の拡大体演算コマンドによる整数型乗算と2の拡大体上乘算との切替が実現される。

【0118】

こうして本実施形態の演算装置では第1の実施形態と同様な動作が実現される。

【0119】

上述したように、本発明の実施の形態に係る演算装置及び暗号処理装置は、整数型乗算回路11、2の拡大体上乘算回路12及びセレクタ13に代えて乗算回路41を用いるようにし、一つの回路41で回路11、12及び13の機能を実現するようにしたので、第1の実施形態と同様な効果が得られる他、より少ない回路追加で整数型乗算と2の拡大体上乘算との切り替えを実現にすることができる。

【0120】

なお、本実施形態ではキャリー制御機能付き全加算器42として図5に示すものを用いるようにしたが、キャリー制御機能付き全加算器42に代えて、図6又は図7に示すキャリー制御機能付き全加算器43又は44を用いるようにしてもよい。

【0121】

(第3の実施形態)

図10は本発明の第3の実施形態に係る演算装置及び暗号処理装置に適用されるコプロセッサの構成例を示すブロック図である。

【0122】

本実施形態は、第1の実施形態に関し、剰余算の部分を具体的に示す具体例であり、図示するように、制御部5において、前述した機能に剰余算機能が付加された2の拡大体演算制御部22aと、この剰余算機能に制御され、且つ逆数計算部51を有する商立て回路50とを備えている。

【0123】

ここで、2の拡大体演算制御部22aは、(5)式の乗算結果 $c'(x)$ を得るために演算部4を制御する前述した機能に加え、この乗算結果 $c'(x)$ に対して法多項式 $f(x)$ による剰余算を実行させるように、演算部4及び商立て回路50を制御する機能をもっている。具体的には、制御機能は、後述する演算アルゴリズムに基づいて、メモリ2やバッファX, Y, Z, Rとの間でデータを入出力する機能と、この入出力に連動し、乗算コマンド、加算コマンド及び逆数計算コマンドなどの各種コマンドを生成して対応する演算回路に与える機能とを有している。

【0124】

商立て回路50は、数式的には剰余算のうちで、被除多項式 $c'(x)$ を法多項式 $f(x)$ で除した商を算出するためのものであり、ここでは、法多項式 $f(x)$ の逆数 $\beta(x)$ と被除多項式 $c'(x)$ とを乗算して前述した商を求める機能をもっている。

【0125】

具体的には、商立て回路50は、2の拡大体演算制御部22aにより制御され、図11に示すように、剰余算の1回目のみメモリ2内の法多項式 $f(x)$ の上位2ブロック($FL-1(x)$, $FL-2(x)$)を逆数計算部51に与えてその上位2ブロックの逆数 $\beta(x)$ を算出させる機能と、得られた $\beta(x)$ がメモリ2に書込まれると、メモリ2からこの逆数を読み出す機能と、読み出した逆数 $\beta(x)$ と現在の被除多項式の上位2ブロック($C'L-1(x)$, $C'L-2(x)$)とを乗算して商 $r(x)$ を得る機能と、得た商 $r(x)$ の上位2ブロック目を商 $qi(x)$ として立て、この商 $qi(x)$ をメモリ2に書込む機能と、これら逆数 $\beta(x)$ の読みから商 $qi(x)$ の書込みまでの動作を剰余 $c(x)$ を得るまで繰り返す機能とをもっている。

【0126】

逆数計算部 51 は、図 12 に示すように、メモリ 2 内の法多項式 $f(x)$ の上位 2 ブロック (FL-1(x), FL-2(x)) を商立て回路 50 から受けると、図 13 に示すように、この 2 ブロック (FL-1(x), FL-2(x)) に基づいてその逆数 $\beta(x)$ を算出する機能と、得られた逆数 $\beta(x)$ をメモリ 2 に書込む機能とをもちている。また、逆数計算部 51 は、その一部に図 27 に示した LFSR が割り算回路として使用されている。

【0127】

ここで、逆数 $\beta(x)$ は、ビット数が固定長であり、後段の除算処理本体にて除数の正規化並びに演算結果の逆正規化の処理を不要とする観点から、単なる逆数ではなく、図 13 に示したように予め補正されている。また、逆数 $\beta(x)$ 自体は、逆数計算部 51 を含む商立て回路 50 に代えて、演算部 4 に算出させてもよい。

【0128】

なお、逆数計算部 51 は、例えば整数型の積和演算回路のバス幅が 8 ビットのように小さいとき、全ての 8 ビット値の逆数値をテーブルにして ROM などに記憶させる方式に置き換えが可能である。しかしながら、逆数計算部 51 は、バス幅が 16 ビット以上のとき、コストを低減させる観点から、全ての 16 ビット値の逆数値を ROM に記憶させる方式よりも好ましい。

【0129】

次に、以上のように構成された演算装置 (コプロセッサ) の動作を説明する。

【0130】

本発明に係る 2 の拡大体多項式基底の剰余乗算は、乗算と剰余算とを別に行う。すなわち、図 14 に示すように、乗算対象の多項式 $a(x)$, $b(x)$ 及び法多項式 $f(x)$ を入力し (ST1)、 $a(x) \cdot b(x)$ の乗算を行って 2 倍のビット長の乗算結果 $C'(x)$ を得た後 (ST2)、 $C'(x) \bmod f(x)$ の剰余算を行い (ST3)、剰余 $c(x)$ を得る (ST4)。

【0131】

ここで、ステップ ST2 の乗算は、第 1 及び第 2 の実施形態で述べた通りであ

る。よって、ここではステップST3～ST4の剰余算の動作について説明する。なお、始めに筆算について延べ、次いで、筆算に対応した実際の処理を説明する。

【0132】

(6) 式の剰余算は、図15に筆算を示すように、除数 $f(x)$ と、被除数 $C(x)$ とが所定ビット数 k の単位ブロック毎に分割されて行われる。なお、単位ブロックのビット数は、例えばコプロセッサ1のバス幅に対応したビット数が適用可能である。

【0133】

次に、被除数 $c(x)$ の上位ブロック $c_{L-i}(x)$ が $f(x)$ で除算され、上位桁から1ブロックの商 $q_i(x)$ が立てられ、 $c(x) - f(x) \cdot q_i(x)$ として、上位桁から1ブロックの被除数 $c(x)$ が減らされる。

【0134】

詳しくは、1ブロックの商 $q_i(x)$ と除多項式 $f(x)$ との乗算結果が毎回 $(m+1)$ ブロックになる。また、この乗算結果を今回の被除多項式 $c(x)$ から減算 (= 加算) して、 $(2m-1 * n)$ ブロックの次回の被除多項式を算出し (n は乗算回数)、すなわち前回の被除数 $c(x)$ を1ブロックずつ減らしていく。

【0135】

剰余算は、このような商立てから減算までの処理を n 回 (= 被除数のビット数 / 単位ブロックのビット数) 繰り返し、剰余 $c(x)$ を得て完了する。

【0136】

続いて、剰余算を行う実際の処理動作を説明する。

上述した剰余算において、商 $q_i(x)$ を立てる動作は、図11に示すように商立て回路50が行い、 $c(x) - f(x) \cdot q_i(x)$ として被除数 $c(x)$ を減らす動作は、図16に示すように演算部4が行う。これら商立て回路50及び演算部4の動作について、以下、順次述べる。

【0137】

商立て回路50は、最初の1回目の商計算を行うとき、図11及び図12に示すように、除数 $f(x)$ の逆数 $\beta(x)$ を算出するため、メモリ2から除数 $f(x)$

) の上位 2 ブロック ($F_{L-1}(x)$, $F_{L-2}(x)$) を読出すと、逆数計算部 51 に入力する。

【0138】

逆数計算部 51 は、図 13 及び (8) 式に示すように、この 2 ブロック ($F_{L-1}(x)$, $F_{L-2}(x)$) のうち、上位 1 ブロック $F_{L-1}(x)$ の最上位ビット MSB から連続した 0 の数を d として記憶する。

【0139】

$$d = \text{count_zero}(F_{L-1}(x)) \quad \dots (8)$$

但し、 $\text{count_zero}()$: () の値で MSB から連続した 0 の数を数える関数

また、この無効な桁数 d に基づいて、後述する左シフトの桁数 h を (9) 式のように算出して記憶する。

【0140】

$$h = (d + 1) \bmod k \quad \dots (9)$$

次に、逆数計算部 51 は、図 13 及び (10) 式に示すように、除数 $f(x)$ の上位 2 ブロック ($F_{L-1}(x)$, $F_{L-2}(x)$) の逆数 $\alpha(x)$ を LFSR 90 にて算出する。

【0141】

$$\alpha(x) = x^{2k} / (F_{L-1}(x) \cdot X^k + F_{L-2}(x)) \quad \dots (10)$$

例えば、1 ブロックが 16 ビット ($k = 16$) の場合を説明する。また、被除数は、最上位ビット MSB が “1” で他のビットが “0” の $x^{2 \cdot 16} (= x^{2k})$ であるとする。

【0142】

逆数計算部 51 は、上位 2 ブロック ($F_{L-1}(x)$, $F_{L-2}(x)$) を除数として図 27 中の係数器 93 に設定した後、被除数 x^{2k} を高次からシフトレジスタに入力して 1 クロック毎のシフトを $2 \cdot 16$ 回繰り返して、32 ビットの逆数 $\alpha(x)$ を得る。なお、1 ブロックは、8 ビットや 32 ビットでもよく、その他の任意のビット数でも同じ方式で逆数 $\alpha(x)$ を算出可能となっている。

【0143】

続いて、逆数計算部 51 は、この逆数 $\alpha(x)$ をその MSB 側に $(k - 1)$ ビ

ットの“0”と1ビットの“1”を連結して2kビットの値 $\alpha'(x)$ にする。
その後、この2kビットの値 $\alpha'(x)$ を、(11)式に示すように、(9)式
で得た左シフトの桁数hだけ左にビットシフトし、補正された逆数 $\beta(x)$ を算
出する。

【0144】

$$\beta(x) = \alpha'(x) \cdot x^h \quad \dots (11)$$

ここで、補正された逆数 $\beta(x)$ とは、以上の(8)式～(11)式を満たす値
である。また、逆数 $\beta(x)$ は、与えられた法多項式 $f(x)$ に対して1回だけ算
出してメモリ2に保持し、以後はメモリ2から読み出される。また、被除数が変
わっても、法多項式 $f(x)$ が同一であれば逆数 $\beta(x)$ も同一であるため、逆数
 $\beta(x)$ を算出せずにメモリ2から読出せばよい。

【0145】

商計算のうち、逆数 $\beta(x)$ が事前設定されていると、剰余算は、以下の(12)
式～(15)式で実行できる。

【0146】

すなわち、商立て回路50は、(12)式及び図11に示すように、現在の被
除数 C_i ($0 \leq i \leq n$)の上位2ブロック($CL-1(x)$, $CL-2(x)$)と逆数 $\beta(x)$
を掛ける。

$$\gamma(x) = \beta(x) \cdot (CL-1(x) \cdot x^k + CL-2(x)) \quad \dots (12)$$

また、商立て回路50は、(13)式に示すように、この結果 $\gamma(x)$ のうち、
1ブロック分の商 $q_i(x)$ に該当する桁を上位2ブロック目として切出し処理し、
メモリ2に書込む。

$$q_i(x) = \gamma(x) / x^{2k} \quad \dots (13)$$

これにより、1ブロック分の商 $q_i(x)$ が得られる。

次に、現在の被除数 $c_i(x)$ から除数と商の積 $f(x) \cdot q_i(x)$ を減算する処理は
、図16に示すように、演算部4が実行する。

【0147】

すなわち、演算部4では、2の拡大体上乘算回路12が、法多項式 $f(x)$ と、
現在の1ブロック分の商 $q_i(x)$ とを乗算して(14)式に示すように乗算結果P

(x) を得る。

【0148】

$$P(x) = f(x) \cdot q_i(x) \cdots (14)$$

また(15)式に示すように、加算回路14、15が、この乗算結果 $P(x)$ を現在の被除数 C_i から減算し(=加算し)、次回の被除数 C_{i+1} を得る。

$$C_{i+1} = C_i + P(x) \cdots (15)$$

以下、(12)式～(15)式を n 回、繰り返して行い、最終的に、図14～図16に示す如き、剰余 $c(x)$ を得る。この剰余 $c(x)$ (= $[c_{m-1}, \dots, c_1, c_0]$)が、(3)式に示した最終的な剰余乗算結果 $c(x)$ に相当する。

【0149】

以上により、第1又は第2の実施形態に述べた(5)式の乗算結果 $c(x)$ から(6)式の剰余算結果 $c(x)$ を算出でき、もって、乗算及び剰余算からなる剰余乗算を完結することができる。

【0150】

(評価)

続いて、以上のように剰余乗算を行う第1及び第3の実施形態におけるコプロセッサ1の処理速度と回路規模を評価したので、順次説明する。

(処理速度の評価)

コプロセッサ1における各コマンドの所要クロック数は、ビット数 $m=160$ の場合と $m=1024$ の場合に関し、図17に示す通りである。なお、楕円曲線暗号への応用では、ビット数 $m=160$ が典型的なサイズである。図示した $m=1024$ の場合は整数型RSA方式暗号の現在安全であるとされる最大の鍵長が1024ビットであることから、将来想定される楕円曲線暗号への鍵長の増加に伴う速度の見積もりとして示した。

【0151】

次に、処理速度の相対比較のため、160ビットの2の拡大体 $GF(2^{160})$ の加算、乗算、二乗算における処理クロック数を評価した。結果は図18に示す通りである。なお、乗算と二乗算と乗算のクロック数は、 $GF(2^{160})$ 演算の速度比較のため、図17とは異なり、法多項式による剰余算を含むクロック数と

なっている。

【0152】

また、相対比較値としてのSR比は、コプロセッサ1のクロック数を通常のシフトレジスタ回路のクロック数で除した値であり、この値が小さいほど処理速度が高速であることを示す。このSR比によれば、本発明のコプロセッサ1は、加算を除き、通常のシフトレジスタ回路と同等の処理速度で2の拡大体演算を実行できることが分かる。

【0153】

(回路規模の評価)

コプロセッサ1の回路規模は、図19に示すように、全体で約30kゲートの規模となる。このコプロセッサ1の回路は、整数型コプロセッサに対し、2の拡大体演算を処理するための回路を追加したものである。

【0154】

具体的には図20に示すように、演算部4では、積和演算回路において、キャリー有り・無し切替回路を追加している。制御部5では、加算、乗算及び二乗算に関し、追加がほぼ不要であるが、除算に関し、商立て回路50を追加している。RAM(メモリ2)及びI/Fは、整数型コプロセッサと共用するため追加が不要である。

【0155】

これにより、追加回路の規模は、全体で約5kゲートとなる。5kゲートという追加回路量は、最近のLSI技術では大きい量ではなく、現行コプロセッサに代えて、本発明のコプロセッサ1を用いることが十分可能な範囲にある。

【0156】

また比較のため、本発明のコプロセッサ1を用いずに2の拡大体の演算機能(加算・乗算・二乗算)を実現する場合における2の拡大体演算専用のコプロセッサの回路規模を見積もった。結果は、図21に示す通りである。

【0157】

図示するように、2の拡大体演算専用のコプロセッサの回路規模は、 $m=160$ の場合に10kゲートであり、 $m=1024$ の場合には16kゲートである。

これにより、2の拡大体の演算機能を実現する場合、本発明に係るコプロセッサ1を用いた方が、2の拡大体演算専用のコプロセッサを設ける場合に比べ、約 $1/2 \sim 1/3$ という少ない追加回路で実現できることが分かる。

【0158】

上述したように本実施形態によれば、第1の実施形態の効果に加え、剰余算処理の際に線形フィードバックシフトレジスタLFSR90ではなく、多倍長の積和演算回路が演算を行うので、1以上の任意の拡大次数 m を使用でき、2の拡大体の拡大次数 m を増加しても、装置本体を作り直さずに演算を実行することができる。さらに、拡大次数 m の制限によるハードウェア的な制約を無くすことで暗号鍵のビットの増加にも対応することができる。

【0159】

また、2の拡大体の剰余乗算を乗算処理と剰余算（除算）処理とに分割し、任意の法多項式 $f(x)$ を使用できるようにしたので、汎用性を向上させることができる。

【0160】

また、商立て回路50が、剰余算処理のとき、被除多項式 $c(x)$ と除多項式 $f(x)$ とに基づいて商計算を行い、上位からバス幅と同じビット数の1ブロックの商多項式 $q_i(x)$ を立てると、演算部4が、この商多項式 $q_i(x)$ と除多項式 $f(x)$ との乗算結果 $q_i(x) \cdot f(x)$ を今回の被除多項式 $c_i(x)$ から減らして次回の被除多項式 $c_{i-1}(x)$ を算出する。

【0161】

コプロセッサ1は、このような商立て回路50による商計算と、演算部4の積和演算による被除多項式データの算出までの処理を繰り返して剰余 $c(x)$ を得るので、ハードウェアの特性を生かして剰余算と商計算の効率化を図ることができる。

【0162】

さらに、商立て回路50としては、商計算のとき、除多項式データの上位2ブロックの逆数データと今回の被除多項式データの上位2ブロックとを乗算し、この乗算結果の上位2ブロック目を1ブロックの商多項式データとすることにより

、得られた商多項式のうち、有効数字の部分を抽出できるので、演算精度の最適化を図ることができる。

【0163】

また、商計算において、除多項式 $f(x)$ の上位2ブロックからの逆数 $\beta(x)$ の算出をコマンドとして独立させ、2の拡大体演算に先行して逆数 $\beta(x)$ を算出し、得られた $\beta(x)$ をメモリ2に格納し、剰余算の実行時には逆数 $\beta(x)$ をメモリ2から読出す。

【0164】

すなわち、同じ法多項式下での重複する剰余算の実行時に、逆数データをメモリから読出して商を立てるので、2回目の商計算以降は逆数データの算出時間を省略でき、もって、2の拡大体乗算（剰余乗算）、二乗算の処理時間を短縮することができる。また、逆数 $\beta(x)$ を事前に算出できるので、2の拡大体の剰余乗算を、乗算と加算とを行う積和演算回路のみを用いて実現することができる。

【0165】

次に、商立て回路50としては、逆数データを算出するとき、除多項式データの上位2ブロックのうち、上位から連続した0の数を計数すると、上位から1ブロック+1ビットの多項式データを最上位ビットを1とするように抽出し、この抽出した多項式データの逆数を求め、得られた逆数の最上位ビット側に、最下位ビットが1で他のビットが0の1ブロックの補正データを連結して全体で2ブロックのデータを求め、このデータを前記計数した0の数だけ上位側にビットシフトさせた結果を逆数データとする。

【0166】

このため、一般的な多倍長の整数型の除算で用いられる単精度除算を用いたKnuthのアルゴリズムによる除数の正規化や近似商の補正処理、商や剰余といった演算結果の逆正規化の処理を行わないように、予め補正した値を逆数データとしたので、ビットシフトの回数を減らすことができ、演算装置を最適化することができる。

【0167】

例えば、整数型の乗算では、 m ビット* m ビット= $2m$ ビットとなり、 $2m$ ビ



ットの上位に数ビットの連続した 0 が並んでいる場合でも有効なビット数は $2m$ である。この乗算結果を用いて除算（剰余算）を行う場合、0 での割り算はできないため、予め除数、被除数を左シフトして MSB に 1 が立つように正規化する必要がある。所定のループを終えて演算が終了した時点で、事前に左シフトしたビット数だけ演算結果（商、剰余）を右シフトする逆正規化の処理も必要である。

一方、本実施形態では、このような除算ループの前後処理を不要にする観点から、商計算時の除数（逆数データ $\beta(x)$ ）を補正したので、演算装置を最適化することができる。

【0168】

また、本実施形態は、ビット単位ではなく、ブロック単位で演算を実行し、且つ補正された逆数 $\beta(x)$ を用いて演算を実行するので、ビットシフトの回数を低減でき、処理速度の高速化を図ることができる。

【0169】

さらに、少ないコマンドと、多倍長積和演算回路を用いた演算方式により、通常のシフトレジスタ型の 2 の拡大体乗算回路と同等の処理速度を有し、且つ、整数型演算や 2 の拡大体演算に基づく各種暗号方式を実行可能な LSI を搭載した演算装置及び暗復号装置を少量の追加回路により実現することができる。なお、2 の拡大体演算を用いた暗号方式としては、素体版楕円番号、多項式基底楕円暗号等の楕円曲線暗号が適用可能となっている。

【0170】

また、本実施形態は、第 1 の実施形態における除算過程の具体例として説明したが、第 2 の実施形態における除算過程の具体例としても、同様の作用効果を得ることができる。

【0171】

（第 4 の実施形態）

図 22 は本発明の第 4 の実施形態に係る演算装置及び暗号処理装置に適用されるコプロセッサの構成例を示す模式図である。

【0172】

本実施形態は、第1～第3の実施形態の変形形態であり、2の拡大体演算専用の演算装置としたものであって、具体的には、整数型乗算器11、セクタ13、整数演算制御部21を省略した構成となっている。但し、演算アルゴリズムは、前述した通りであり、2の拡大体の乗算処理を乗算と剰余算とに分割し、乗算の後に剰余算を実行する。

【0173】

以上のような構成としても、整数型演算自体の作用効果と、整数型演算並びに2の拡大体演算の切替に関する作用効果とを除き、第1～第3の実施形態の効果を得ることができる。換言すると、2の拡大体演算に関し、第1～第3の実施形態の効果を得ることができる。

【0174】

その他、本発明はその要旨を逸脱しない範囲で種々変形して実施できる。

【0175】

【発明の効果】

以上詳記したように本発明によれば、最小のアーキテクチャを追加するだけで整数型の演算に加えて2の拡大体上の演算をも実行できる演算装置及び暗号処理装置を提供することができる。

【0176】

また、2の拡大体の拡大次数 m を増加しても、装置本体を作り直さずに演算を実行できる演算装置及び暗号処理装置を提供できる。

【図面の簡単な説明】

【図1】

本発明の第1の実施形態に係る演算装置の構成例を示すブロック図。

【図2】

$c'(x) = a(x) * b(x)$ を実現するための4*4ビットの単位乗算の回路構成例を示す図。

【図3】

整数型乗算を実現するための4*4ビットの単位乗算の回路構成例を示す図。

【図4】

同実施形態におけるコプロセッサに用いられるキャリー制御機能付きの4ビットのリップルキャリー型全加算器の構成例を示すブロック図。

【図5】

同実施形態の加算回路に用いられる全加算器及びキャリー制御スイッチの構成例を示す図。

【図6】

キャリー制御機能付き全加算器の変形例を示す図。

【図7】

キャリー制御機能付き全加算器の他の変形例を示す図。

【図8】

本発明の第2の実施形態に係る演算装置の構成例を示すブロック図。

【図9】

同実施形態の乗算回路を実現するための4*4ビットの単位乗算の回路構成例を示す図。

【図10】

本発明の第3の実施形態に係る演算装置及び暗号処理装置に適用されるコプロセッサの構成例を示すブロック図

【図11】

同実施形態における商立て回路の構成を示す模式図

【図12】

同実施形態における逆数計算部の機能を説明するための模式図

【図13】

同実施形態における逆数計算部の構成を示す模式図

【図14】

同実施形態における2の拡大体多項式基底の剰余乗算方式を説明するためのフローチャート

【図15】

同実施形態における剰余算処理を説明するための筆算の模式図

【図16】

同実施形態における演算部の処理を示す模式図

【図 1 7】

同実施形態における各コマンドの所要クロック数を示す図

【図 1 8】

同実施形態における $GF(2^{160})$ 演算の所要クロック数を示す図

【図 1 9】

同実施形態におけるコプロセッサの回路規模を示す図

【図 2 0】

同実施形態における追加回路量を示す図

【図 2 1】

同実施形態における比較用の $GF(2^m)$ 演算専用のコプロセッサの回路規模を示す図

【図 2 2】

本発明の第 4 の実施形態に係る演算装置及び暗号処理装置に適用されるコプロセッサの構成例を示す模式図

【図 2 3】

暗号処理用演算用コプロセッサを含む IC カード用 LSI のブロック図。

【図 2 4】

図 1 0 に示す LSI におけるコプロセッサ部分の構成例を示すブロック図。

【図 2 5】

多項式基底の 2 の拡大体演算を行うコプロセッサのハードウェア構成例を示すブロック図。

【図 2 6】

円分体といわれる 2 の拡大体の乗算回路を示す図。

【図 2 7】

一般的な線形フィードバックシフトレジスタ LFSR の構成を示すブロック図

【符号の説明】

1, 1' …コプロセッサ

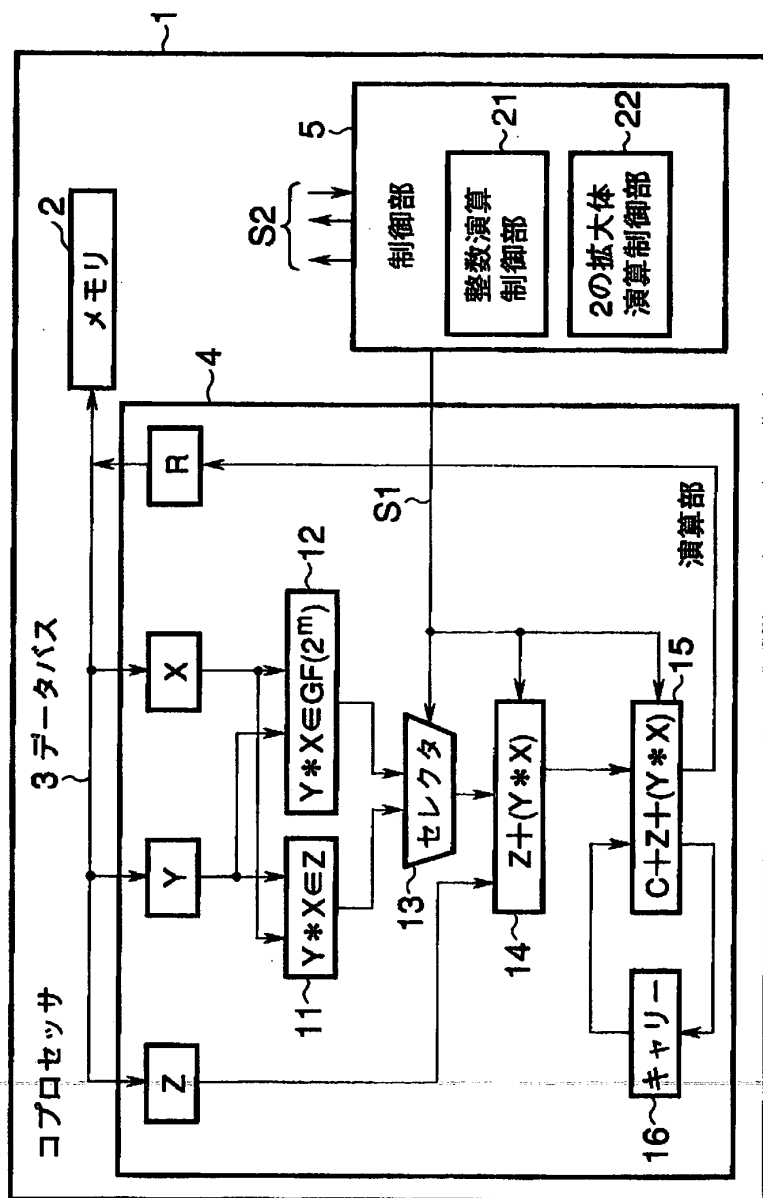
2 …メモリ

- 3…データバス
- 11…整数型乗算回路
- 12…2の拡大体上乘算回路
- 13…セレクタ
- 14…加算回路
- 15…加算回路
- 16…キャリー保持部
- 21…整数演算制御部
- 22…2の拡大体演算制御部
- 32…全加算器
- 33…スイッチ
- 41…乗算回路
- 42, 43, 44…キャリー制御機能付き全加算器
- 50…商立て回路
- 51…逆数計算部
- S1, s2…制御信号

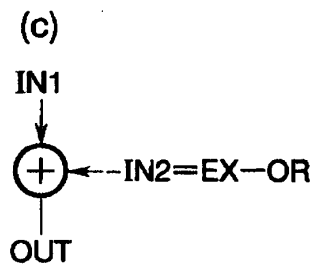
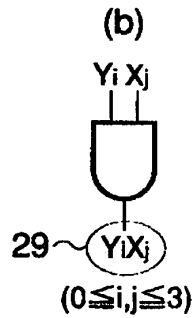
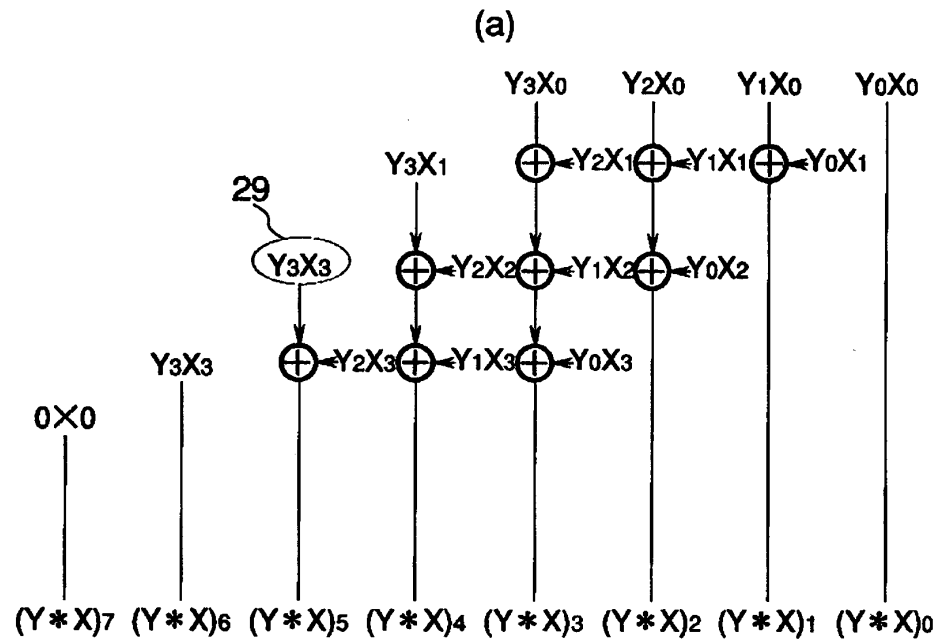
【書類名】

図面

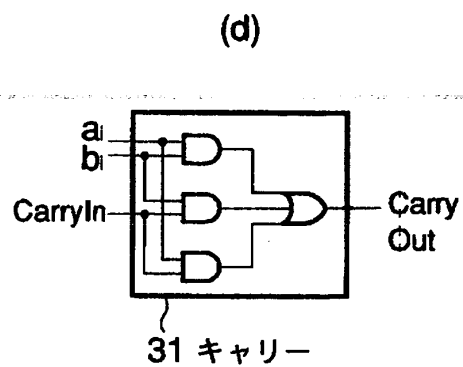
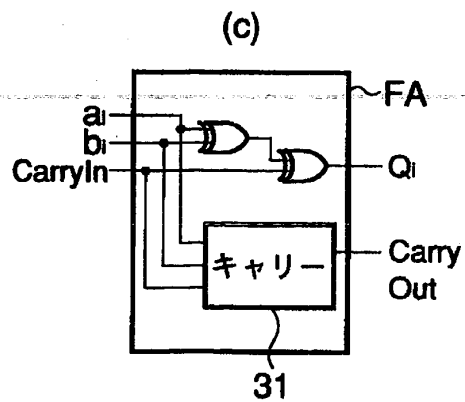
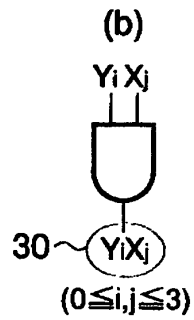
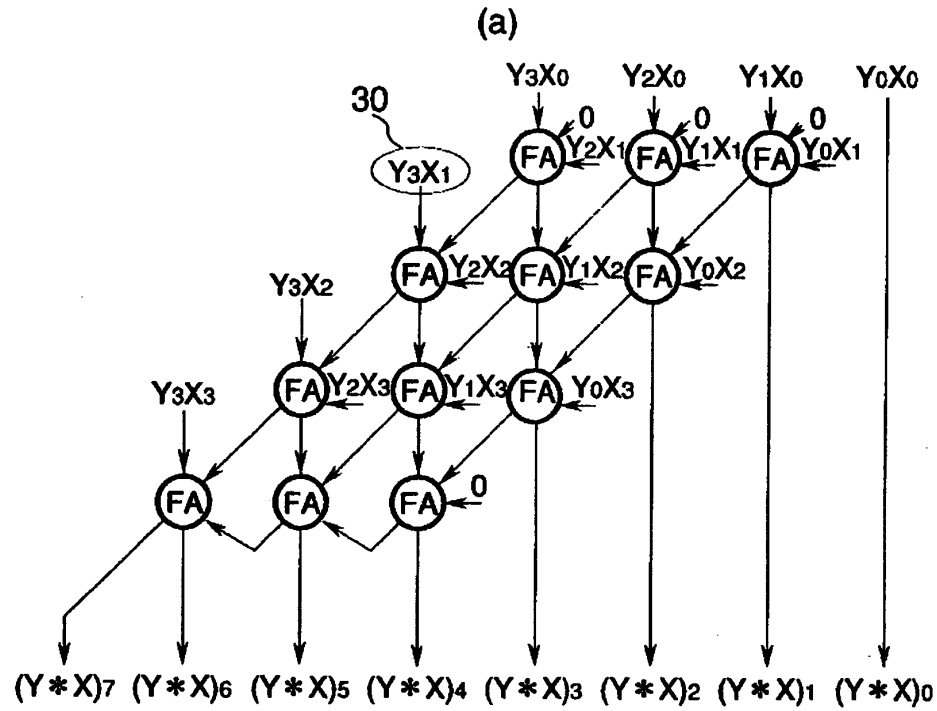
【図 1】



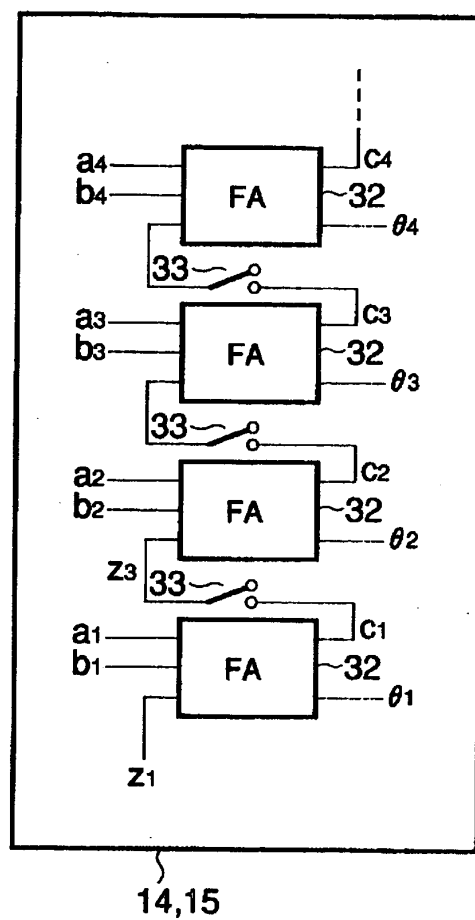
【図 2】



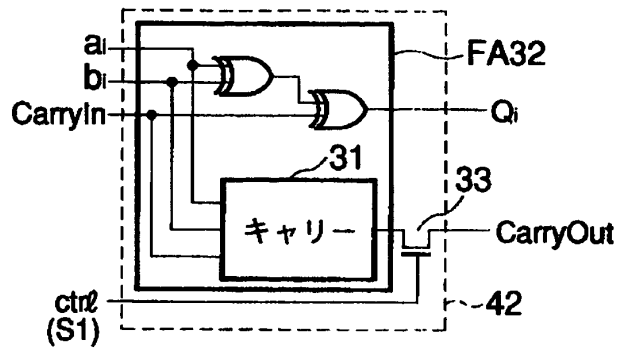
【図 3】



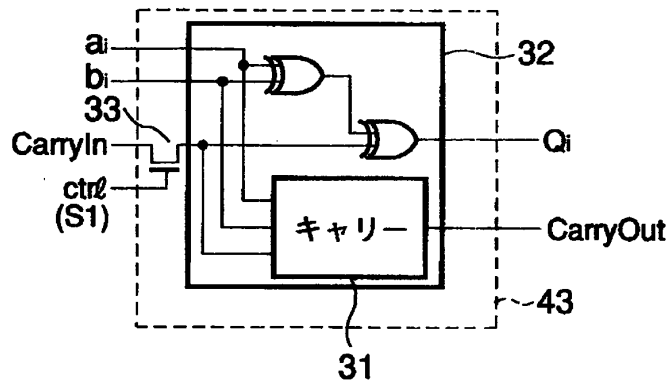
【図 4】



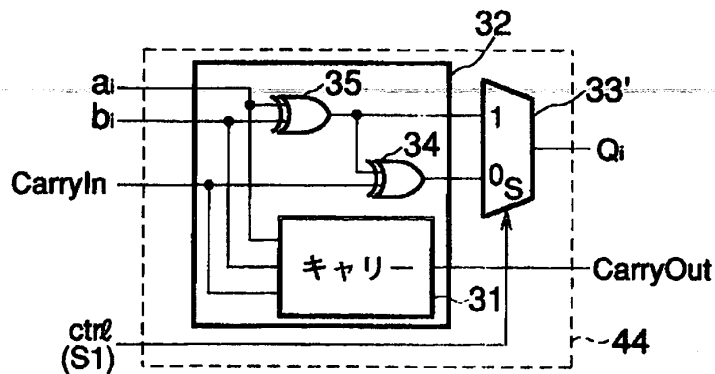
【図 5】



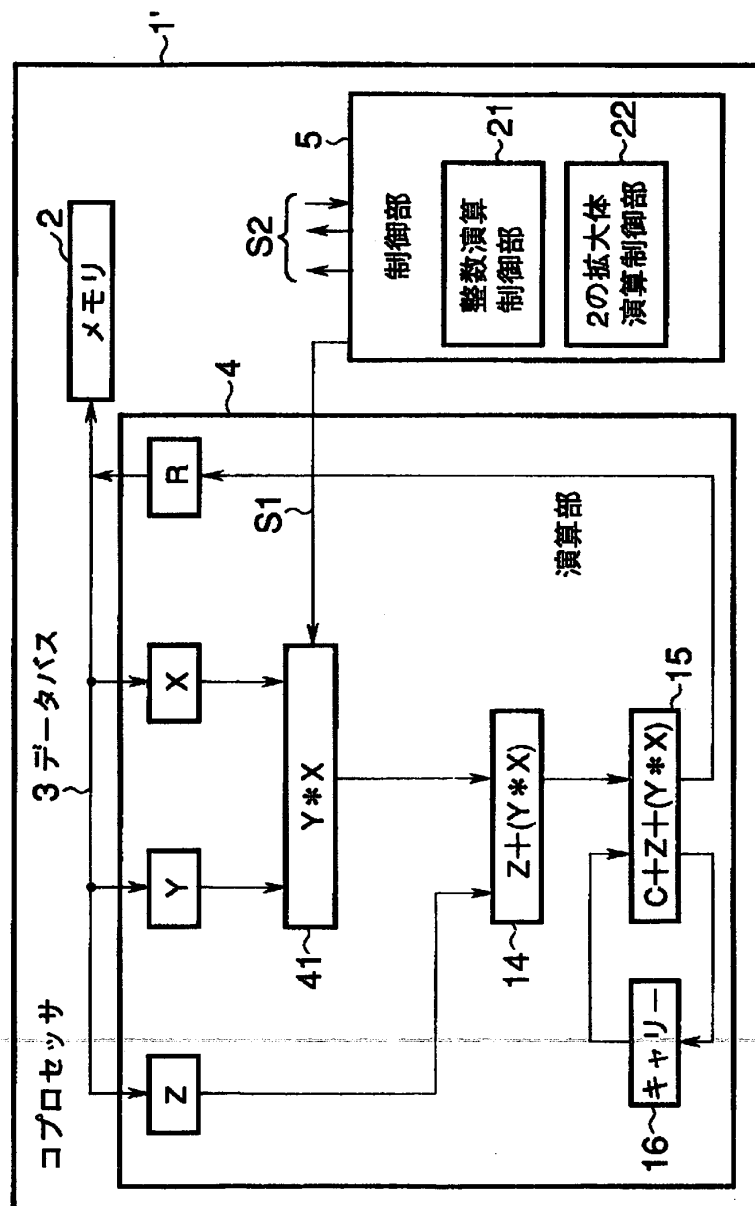
【図 6】



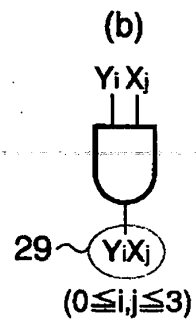
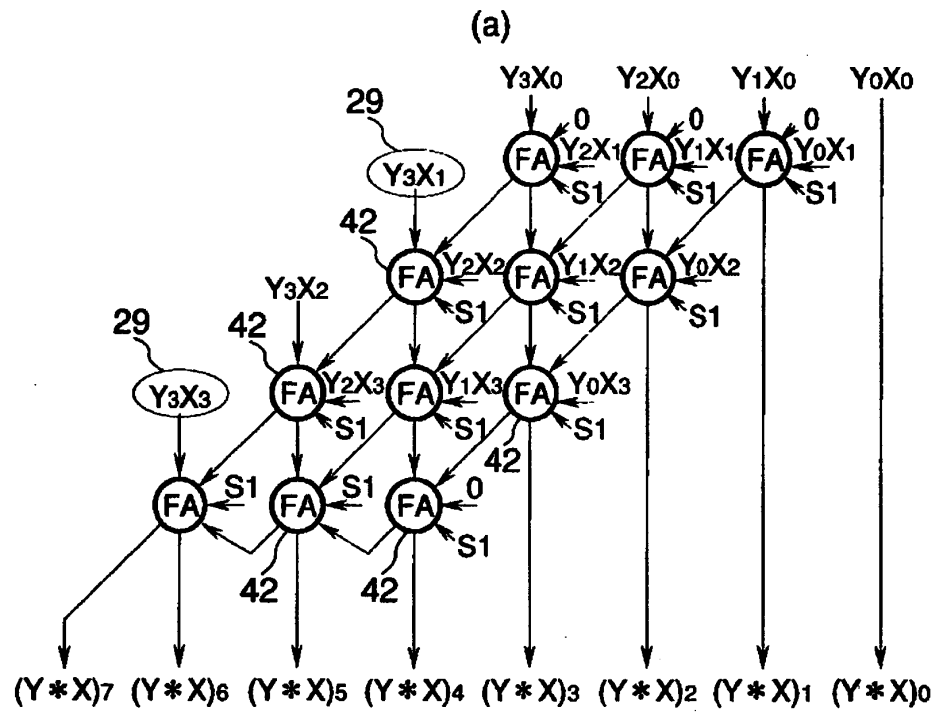
【図 7】



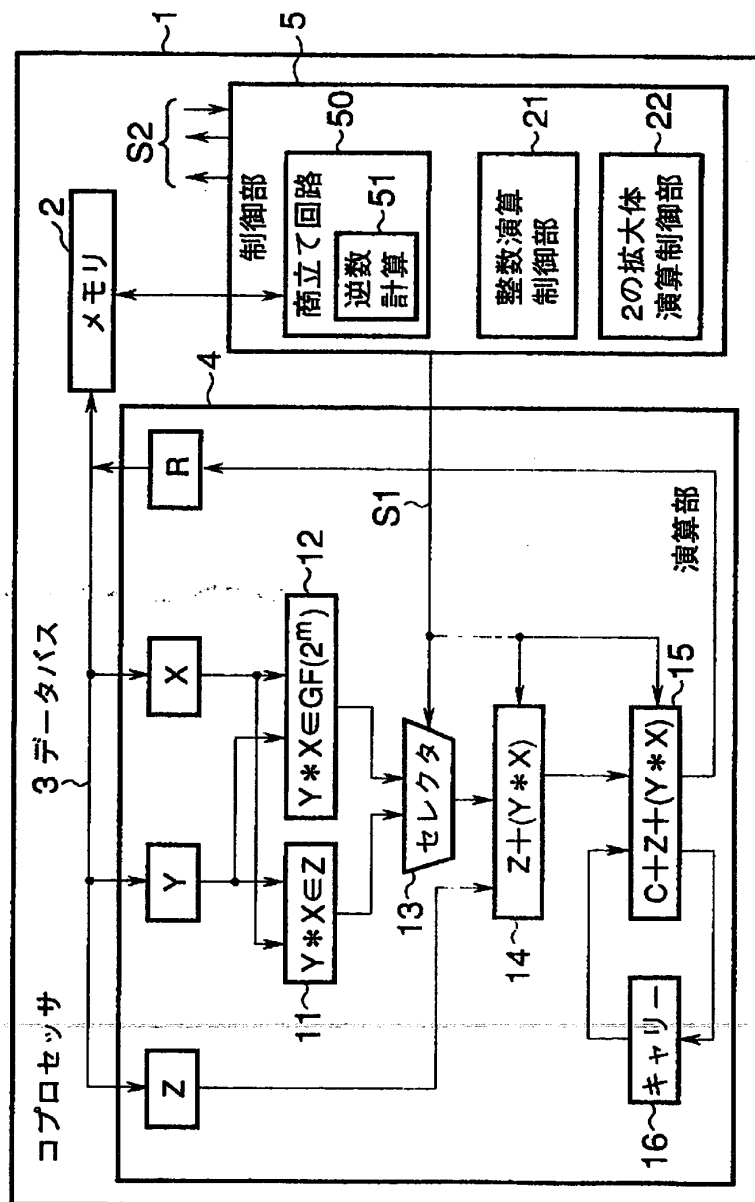
【図 8】



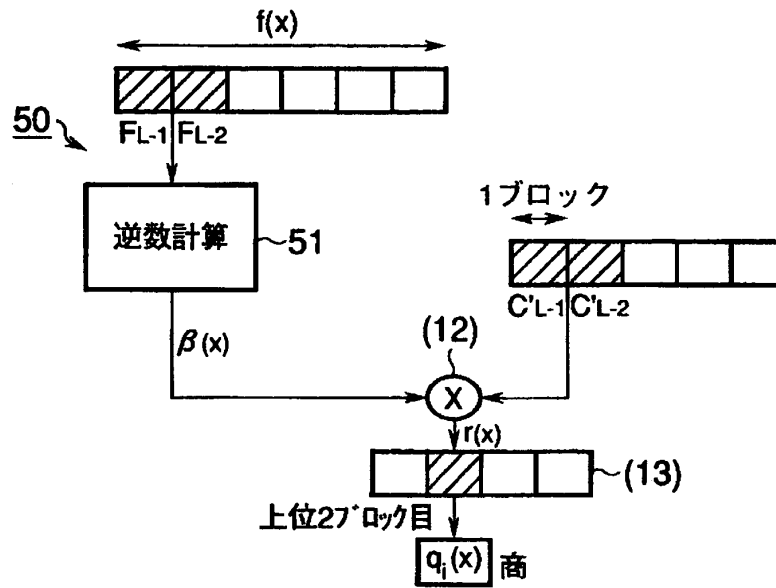
【図 9】



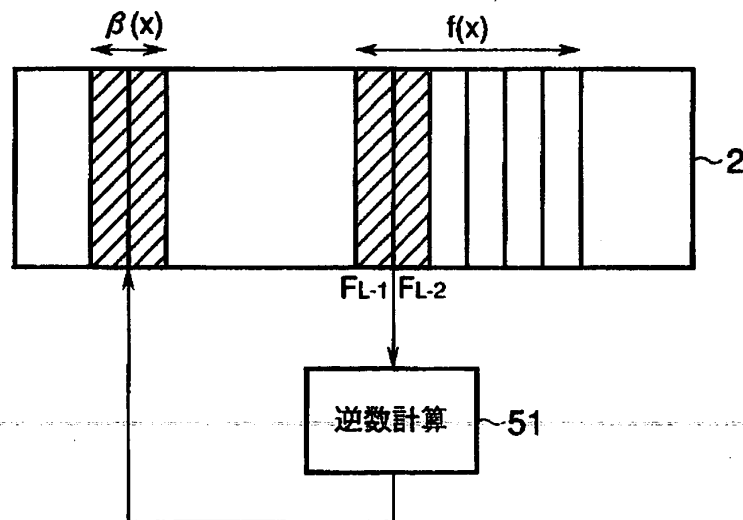
【図10】



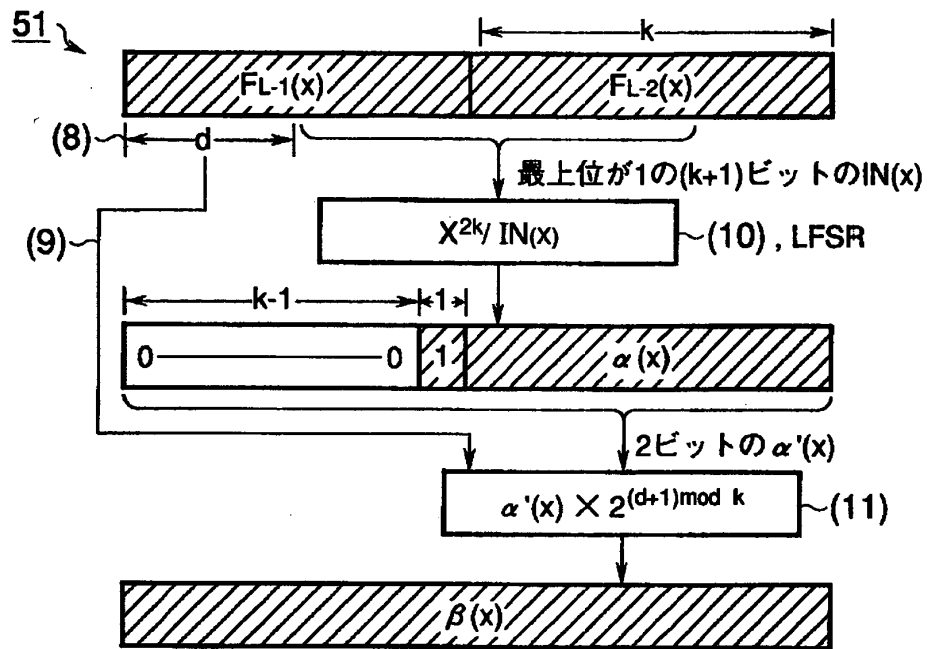
【図 11】



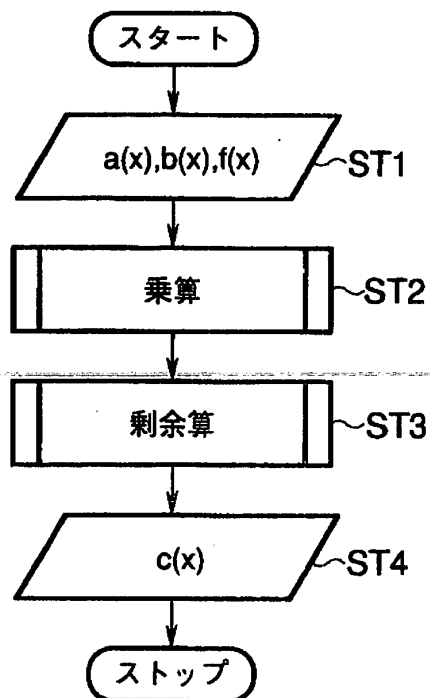
【図 12】



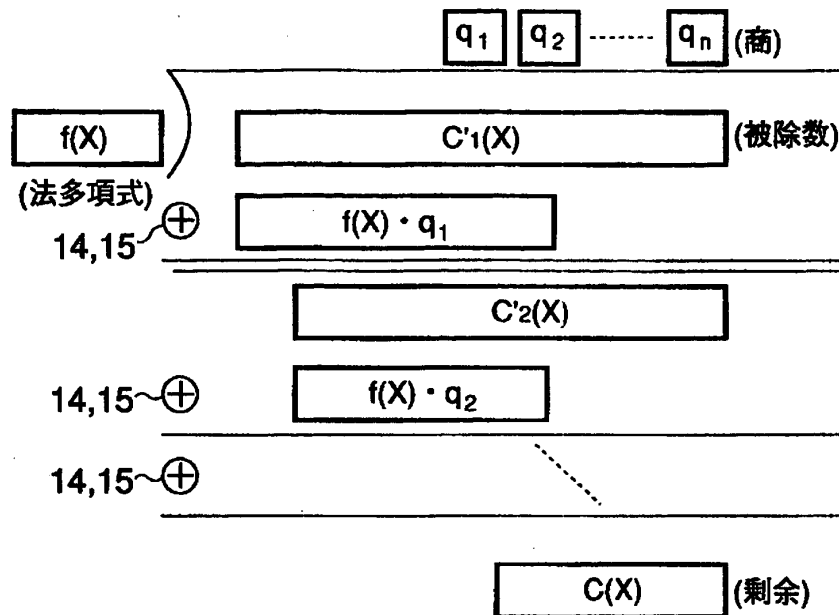
【図 13】



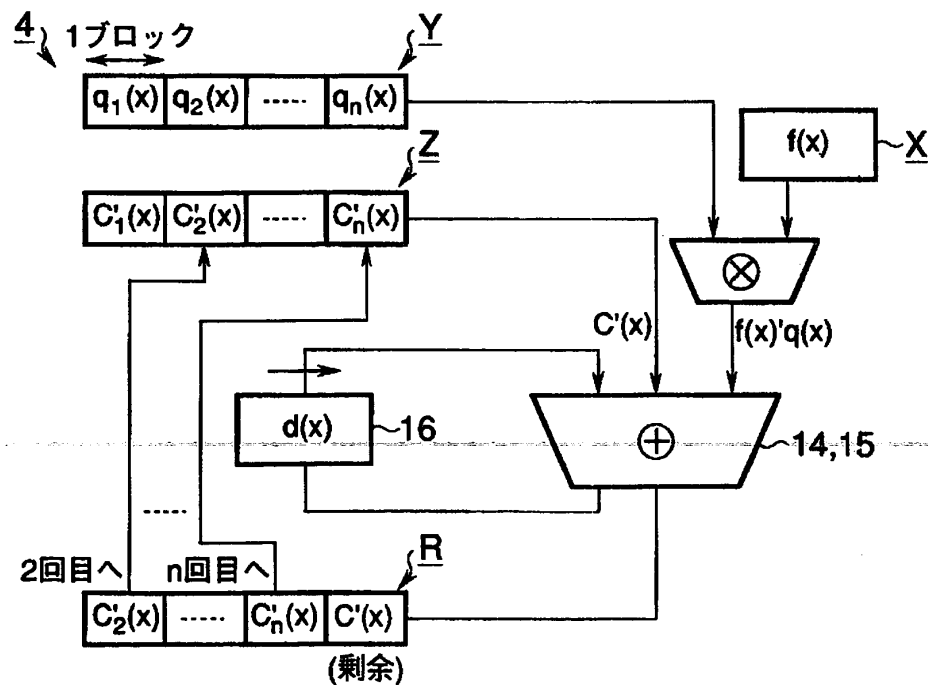
【図 14】



【図 1 5】



【図 1 6】



【図 17】

各コマンドの所要クロック数

コマンド		m=160	m=1024
加算		14	68
乗算		64	2,116
二乗算		25	133
除算	事前計算	35	35
	本体	134	2,564

【図 18】

GF(2¹⁶⁰)の所要クロック数

演算	クロック数	SR比
加算	14	約4.6倍
乗算	198	約1.2倍
二乗算	159	約1倍

(SR比)=(クロック数)/(シフトレジスタ回路でのクロック数)

【図 19】

本発明のコプロセッサ回路規模(ゲート数)

演算部	8k
制御部	12.8k
RAM	8.5k
I/F	0.5k
全体	約30k

【図 2 0】

整数型のコプロへの追加回路量(ゲート数)

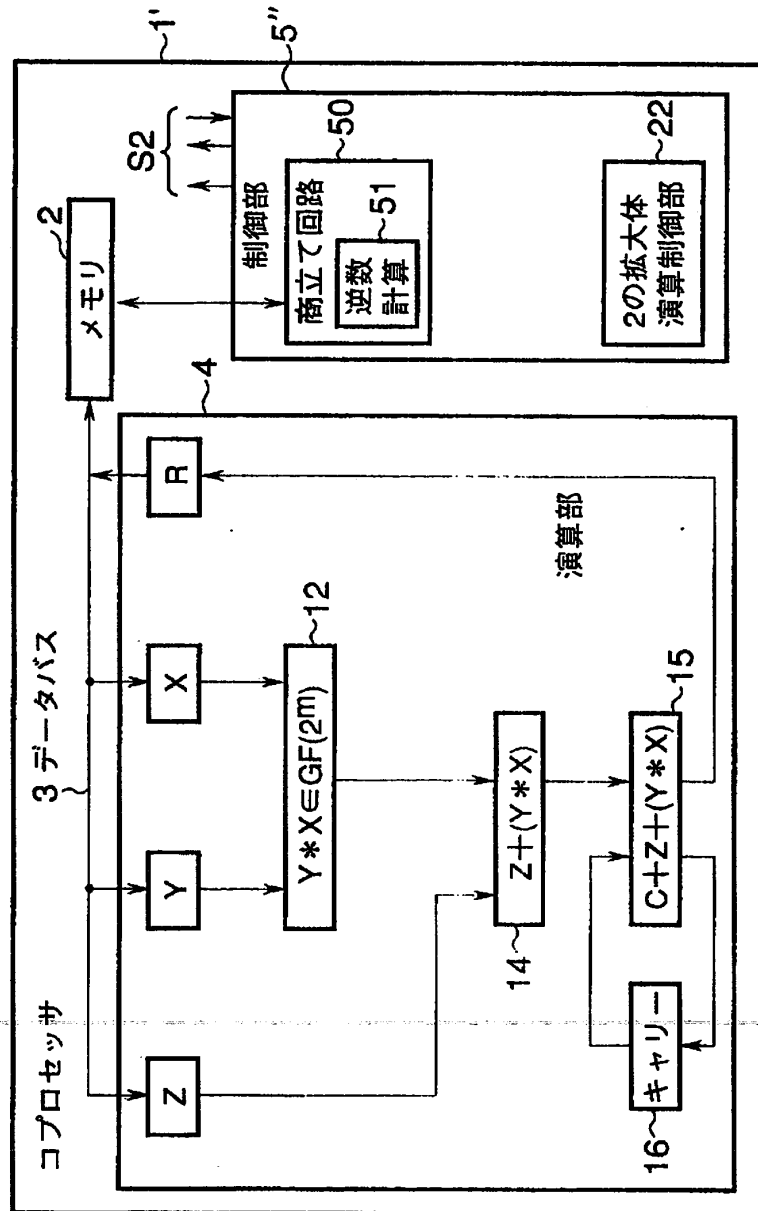
演算部	1k
制御部	3.8k
RAM	0(共用)
I/F	0(共用)
全体	4.8k

【図 2 1】

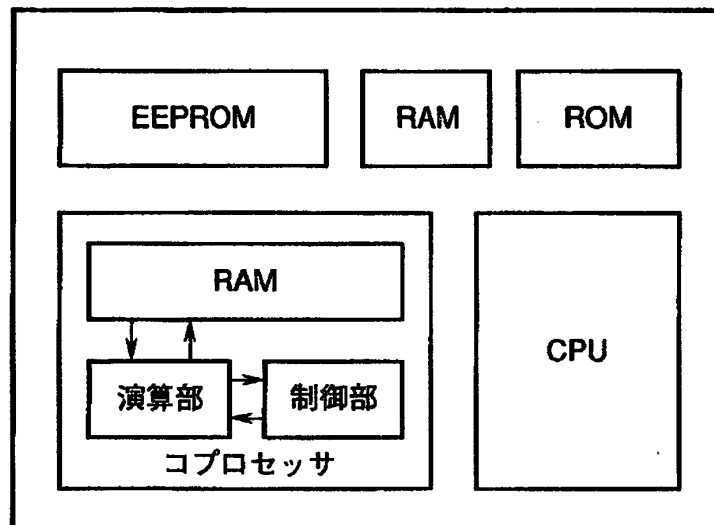
GF(2^m)単独回路規模(ゲート数)

	m=160	m=1024
演算部	3.1k	3.1k
制御部	3.8k	3.8k
RAM	2.3k	8.5k
I/F	0.5k	0.5k
全体	約10k	約16k

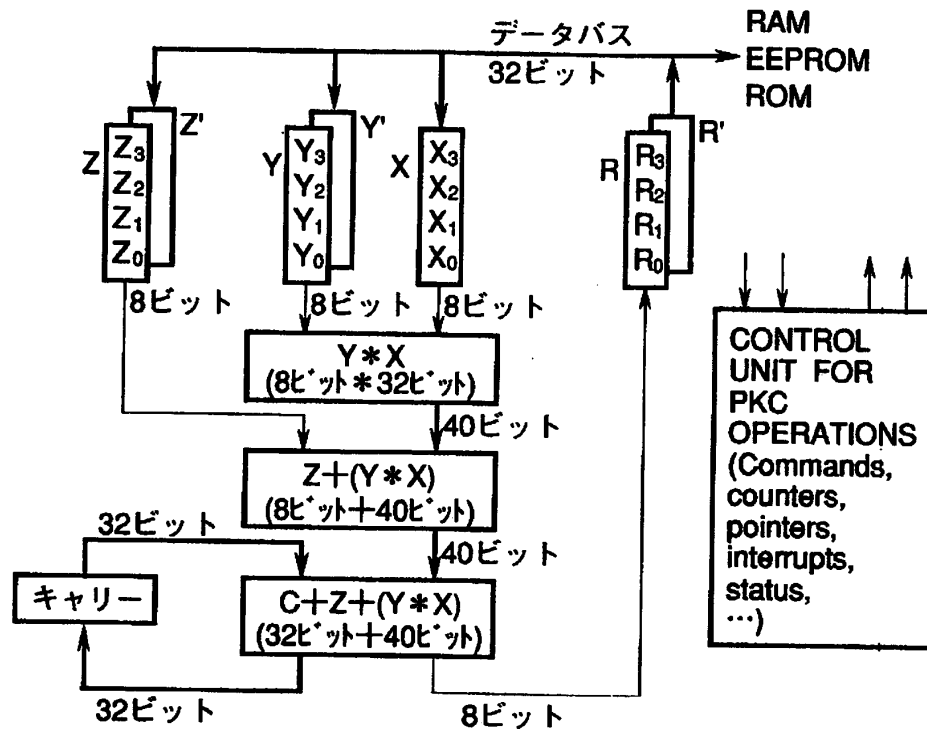
【図 22】



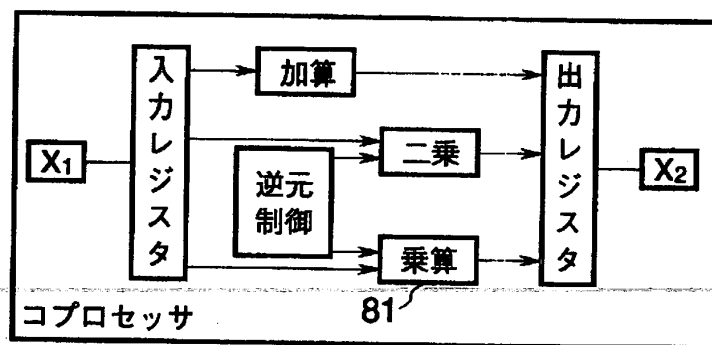
【図 2 3】



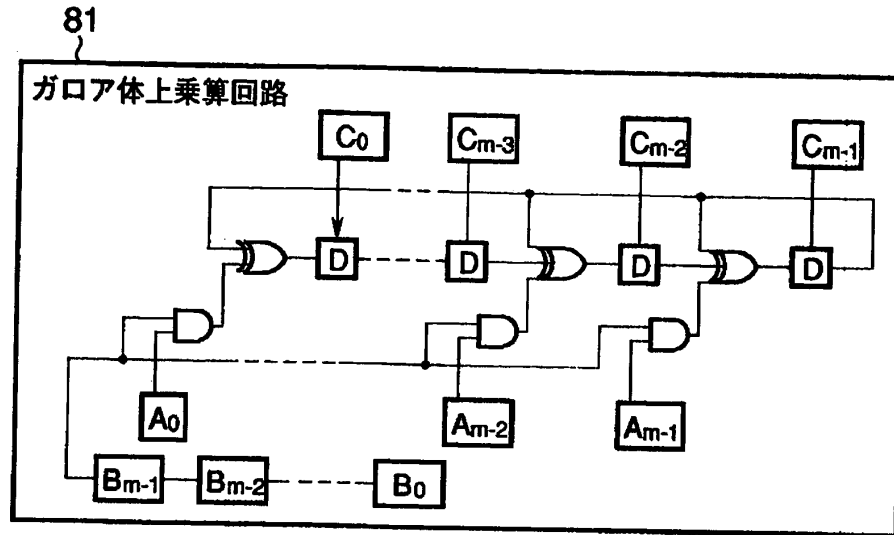
【図 24】



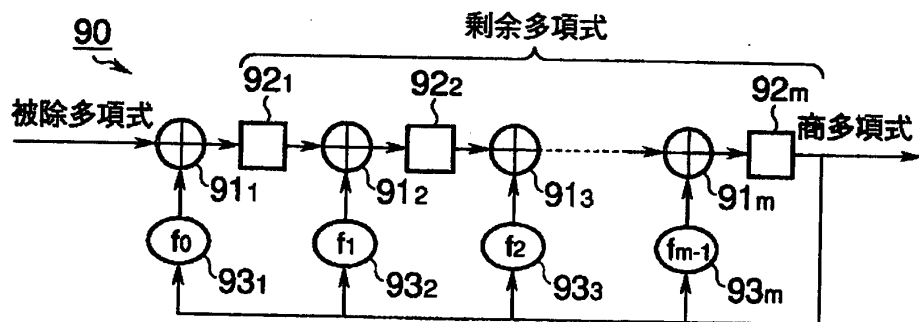
【図 25】



【図 26】



【図 27】



【書類名】 要約書

【要約】

【課題】 本発明は、2の拡大体の拡大次数 m を増加しても、装置本体を作り直さずに演算の実行を図る。

【解決手段】 2の拡大体の多項式基底表現での剰余乗算を実行可能な多倍長の積和演算回路(12, 14, 15)を有し、さらに、剰余乗算を乗算処理と剰余算処理とに分割して積和演算回路の制御により実行するための制御手段(5)を備えた演算装置及び暗号処理装置。

【選択図】 図1

特平11-209831

出 願 人 履 歴 情 報

識別番号

[000003078]

1. 変更年月日

1990年 8月22日

[変更理由]

新規登録

住 所

神奈川県川崎市幸区堀川町72番地

氏 名

株式会社東芝